IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

LIBERTY PATENTS, LLC, Plaintiff, v. LENOVO GROUP LTD.; LENOVO (SHANGHAI) ELECTRONICS TECHNOLOGY CO., LTD.; LENOVO INFORMATION PRODUCTS (SHENZHEN) CO. LTD.; LCFC (HEFEI) ELECTRONICS TECHNOLOGY CO., LTD. D/B/A LC FUTURE CENTER and LENOVO COMPAL FUTURE CENTER; COMPAL ELECTRONICS, INC.; COMPAL INFORMATION TECHNOLOGY (KUNSHAN) CO. LTD. Defendants.	CIVIL ACTION NO. 2:20-cv-314 ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT **JURY TRIAL DEMANDED**

ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Liberty Patents, LLC ("Liberty Patents" or "Plaintiff") files this original complaint against Defendants Lenovo Group Limited, Lenovo (Shanghai) Electronics Technology Co. Ltd., Lenovo Information Products (Shenzhen) Co. Ltd., LCFC (Hefei) Electronics Technology Co. Ltd. d/b/a LC Future Center and Lenovo Compal Future Center, Compal Electronics, Inc., and Compal Information Technology (Kunshan) Co. Ltd. (collectively "Defendants"), alleging, based on its own knowledge as to itself and its own actions and based on information and belief as to all other matters, as follows:

PARTIES

1. Liberty Patents is a limited liability company formed under the laws of the State of Texas, with its principal place of business at 2325 Oak Alley, Tyler, Texas 75703.

2. Defendant Lenovo Group Limited is a company organized under the laws of Hong Kong SAR. Lenovo Group Limited has an office at 23rd Floor, Lincoln House, Taikoo Place, 979 King's Road, Quarry Bay, Hong Kong SAR, China. Lenovo Group Limited may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business. In addition, Lenovo Group Limited may be served with process by serving its officer, managing agent, or general agent, Laura Quatela, who is Lenovo Group Limited's Senior Vice President and Chief Legal Officer.

3. Lenovo Group Limited—through its subsidiaries—manufactures and sells personal computers and handheld devices worldwide.

4. Defendant Lenovo (Shanghai) Electronics Technology Co., Ltd. is a company organized under the laws of the People's Republic of China. Lenovo (Shanghai) Electronics Technology Co., Ltd. has an office at No. 68 Building, 199 Fenju Road, Wai Gao Qiao Free Trade Zone, Shanghai, 200131, China. Lenovo (Shanghai) Electronics Technology Co., Ltd. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

5. Lenovo (Shanghai) Electronics Technology Co., Ltd., an indirect subsidiary of Lenovo Group Limited, is involved in the manufacturing and distribution of personal computers, laptop computers, notebook computers, handheld devices, and other related products.

6. Defendant Lenovo Information Products (Shenzhen) Co. Ltd. is a company organized under the laws of the People's Republic of China. Lenovo Information Products (Shenzhen) Co. Ltd. has an office at ISH2 Building, 3 Guanglan Road, Futian Free Trade Zone, Shenzhen, 518038, China. Lenovo Information Products (Shenzhen) Co., Ltd. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

7. Lenovo Information Products (Shenzhen) Co., Ltd., an indirect subsidiary of Lenovo Group Limited, is involved in the manufacturing and distribution of personal computers, laptop computers, notebook computers, handheld devices, and other related products.

8. Defendant LCFC (Hefei) Electronics Technology Co., Ltd. d/b/a LC Future Center and Lenovo Compal Future Center is a company organized under the laws of a foreign jurisdiction. LCFC (Hefei) Electronics Technology Co. Ltd. d/b/a LC Future Center and Lenovo Compal Future Center has an office at 7th Floor, No. 780, Bei'an Rd., Zhongshan District, Taipei City 10491. LCFC (Hefei) Electronics Technology Co., Ltd. d/b/a LC Future Center and Lenovo Compal Future Center may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

9. LCFC (Hefei) Electronics Technology Co., Ltd. d/b/a LC Future Center and Lenovo Compal Future Center, an indirect subsidiary of Lenovo Group Limited, is involved in the manufacturing and distribution of personal computers, laptop computers, notebook computers, handheld devices, and other related products. Over 60% of Lenovo laptops are manufactured by LCFC (Hefei) Electronics Technology Co., Ltd.[1]

10. The Defendants identified in paragraphs 2 through 9 above (collectively, "Lenovo") are companies that together—with their affiliates—comprise one of the world's leading manufacturers of computers and computer-related products. Indeed, Lenovo was the world's largest supplier of personal computers in 2019 (accounting for over 24% of global PC shipments), and the third largest supplier of personal computers in the United States (accounting for 15% of PC sales).[2] Together, the Lenovo Defendants design, manufacture, use, import into the United States, sell, and/or offer for sale in the United States personal computers, laptop computers, notebook computers, tablets, and other related products. Lenovo's devices are marketed, offered for sale, and/or sold throughout the United States, including within this District.

11. The Lenovo Defendants named above and their affiliates are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and

[1] *See* China's Lenovo to Acquire 20% Stake in Compal in First Possible Mainland Investment in Taiwan's Laptop Equipment Industry, Says Report, YICAI GLOBAL (Dec. 8, 2017), www.yicaiglobal.com/news/china-lenovo-to-acquire-20-stake-in-compal-in-first-possible-mainland-investment-in-taiwan-laptop-equipment-industry-says-report.

[2] *See* Gartner Says Worldwide PC Shipments Grew 2.3% in 4Q19 and 0.6% for the Year, GARTNER (Jan. 13, 2020), www.gartner.com/en/newsroom/press-releases/2020-01-13-gartner-says-worldwide-pc-shipments-grew-2-point-3-percent-in-4q19-and-point-6-percent-for-the-year

using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular.

12. The Lenovo Defendants named above and their affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

13. Thus, the Lenovo Defendants named above and their affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

14. Defendant Compal Electronics, Inc. is a company organized under the laws of a foreign jurisdiction. Compal Electronics, Inc. has an office at No. 581, Ruiguang Rd., Neihu District, Taipei City 11492. Compal Electronics, Inc. may also be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

15. Compal Electronics, Inc.—through its subsidiaries—is involved in the manufacture, processing, and trading of laptop and notebook computers, computer monitors, LCD TVs, cellphones, and other electronics.

16. Defendant Compal Information Technology (Kunshan) Co. Ltd. is a company organized under the laws of the People's Republic of China. Compal Information Technology (Kunshan) Co. Ltd. has an office at No. 58, First Avenue, A Zone Kunshan Comprehensive Free Trade Zone, Kunshan, 215300 Jiangsu, China. Compal Information Technology (Kunshan) Co. Ltd. may also be served with process by serving the Texas Secretary of State, 1019 Brazos

Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

17. Compal Information Technology (Kunshan) Co. Ltd., a subsidiary of Compal Electronics, Inc., is involved in the production of notebooks and other electronics.

18. The Defendants identified in paragraphs 14 through 17 above (collectively, "Compal") are companies that together—with their affiliates—comprise "one of the world's leading manufacturers of notebook PCs, smart devices, data center equipment and LCD products" for the world's top brands.[3] Compal's Notebook PC business accounts for 25% of global market share.[4] Compal is an original equipment/design manufacturer (OEM/ODM). The laptop and notebook computers that Compal designs and manufactures are imported into the United States and are marketed, offered for sale, and/or sold throughout the United States, including within this District.[5]

19. The Compal Defendants named above and their affiliates are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular.

---

[3] *See* https://www.compal.com/why-compal/.

[4] *See* 2019 Annual Report, Compal Electronics, Inc., at 4 (May 13, 2020) (hereinafter "Compal's 2019 Annual Report"), https://www.compal.com/mediafiles/sh-meeting/annual-report/%E4%BB%81%E5%AF%B6108%E5%B9%B4%E5%A0%B1-%E8%8B%B1%E6%96%87%E5%AE%9A%E7%A8%BF-1090609.pdf.

[5] Of Compal's total sales, 42.1% are made in the Americas. *See supra* Compal's 2019 Annual Report at 132.

20. The Compal Defendants named above and their affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

21. Thus, the Compal Defendants named above and their affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

22. While about 60% of Lenovo's laptops and notebooks are manufactured by LCFC (Hefei) Electronics Technology Co., Ltd. (named as a defendant above), the remaining 40% of Lenovo's laptops are outsourced to other companies. About 50% to 60% of these outsourced laptops and notebooks are manufactured by Compal Electronics, Inc. and its affiliates.[6]

23. The partnership between Lenovo and Compal began in 2010 when the two set up LCFC (Hefei) Electronics Technology Co., Ltd. d/b/a LC Future Center and Lenovo Compal Future Center as a joint investment. Lenovo held a 51% stake and Compal 49%. Lenovo acquired approximately a 20% stake in Compal Electronics, Inc. a few years ago.[7]

24. The parties to this action are properly joined under 35 U.S.C. § 299 because the right to relief asserted against defendants jointly and severally arises out of the same series of transactions or occurrences relating to the making and using of the same products or processes, including laptops, notebooks, tablets, and desktop PCs bearing the Lenovo brand. Additionally, questions of fact common to all defendants will arise in this action.

---

[6] *See* China's Lenovo to Acquire 20% Stake in Compal in First Possible Mainland Investment in Taiwan's Laptop Equipment Industry, Says Report, YICAI GLOBAL (Dec. 8, 2017), www.yicaiglobal.com/news/china-lenovo-to-acquire-20-stake-in-compal-in-first-possible-mainland-investment-in-taiwan-laptop-equipment-industry-says-report.

[7] *See id*.

JURISDICTION AND VENUE

25. This is an action for infringement of a United States patent arising under 35 U.S.C. §§ 271, 281, and 284–85, among others. This Court has subject matter jurisdiction of the action under 28 U.S.C. § 1331 and § 1338(a).

26. This Court has personal jurisdiction over the Lenovo Defendants pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) Lenovo has done and continues to do business in Texas; (ii) Lenovo has committed and continues to commit acts of patent infringement in the State of Texas, including making, using, offering to sell, and/or selling accused products in Texas, and/or importing accused products into Texas, including by Internet sales and sales via retail and wholesale stores, inducing others to commit acts of patent infringement in Texas, and/or committing a least a portion of any other infringements alleged herein; and (iii) Lenovo regularly places its products within the stream of commerce—directly, through subsidiaries, or through third parties—with the expectation and knowledge that such products, such as laptop and notebook computers, will be shipped to, sold, or used in Texas and elsewhere in the United States. Accordingly, Lenovo has established minimum contacts within Texas and purposefully availed itself of the benefits of Texas, and the exercise of personal jurisdiction over Lenovo would not offend traditional notions of fair play and substantial justice. In addition, or in the alternative, this Court has personal jurisdiction over Lenovo pursuant to Federal Rule of Civil Procedure 4(k)(2).

27. This Court has personal jurisdiction over the Compal Defendants pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) Compal has done and continues to do business in Texas; (ii) Compal manufactures or assembles—directly, through subsidiaries, affiliates, or third parties—products that are and have been offered for sale, sold,

purchased, and/or used within Texas, including without limitation by downstream customers of Compal, including brand manufacturers such as Acer, ASUS, Dell, HP, and Lenovo,[8] and (iii) Compal regularly places its products within the stream of commerce—directly, through subsidiaries, or through third parties—with the expectation and knowledge that such products, such as laptop and notebook computers, will be shipped to, sold, or used in Texas and elsewhere in the United States. Accordingly, Compal has established minimum contacts within Texas and purposefully availed itself of the benefits of Texas, and the exercise of personal jurisdiction over Compal would not offend traditional notions of fair play and substantial justice. In addition, or in the alternative, this Court has personal jurisdiction over Compal pursuant to Federal Rule of Civil Procedure 4(k)(2).

28. Venue is proper as to Defendants Lenovo and Compal. The Lenovo Defendants and Compal Defendants are organized under the laws of foreign jurisdictions. 28 U.S.C. § 1391(c)(3) provides that "a defendant not resident in the United States may be sued in any judicial district, and the joinder of such a defendant shall be disregarded in determining where the action may be brought with respect to other defendants." *See also In re HTC Corp.*, 889 F.3d 1349 (Fed. Cir. 2018).

BACKGROUND

29. The three patents-in-suit cover technology used in computer systems, such as notebook computers, laptop computers, desktop computers, tablets, and other electronic devices. More particularly, the patents-in-suit describe key improvements to electronic devices in the areas of more efficient handling of computer instructions for faster processing, better power

[8] *See supra* Compal's 2019 Annual Report, at 119.

distribution and power management, and a better process for retrieving automatic software updates.

30. U.S. Patent No. 6,535,959 ("the '959 Patent") discloses a processor that includes an instruction cache. The instruction cache is a set-associative cache that comprises multiple blocks. Claim 1 of the '959 patent is directed to a processor that generates a power reduction signal, which indicates whether the subsequent instruction to be executed resides in the same block of the instruction cache as the current instruction that is being executed. This advantageously allows, for example, the processor to read consecutive instructions (or instructions that are in the same block) quickly, without multiple additional steps. The novel system results in a processor with increased operating speed and decreased power consumption.

31. The invention described in the '959 Patent was the result of research conducted by two inventors at Conexant Systems, Inc., which was—at the time—the world's largest, standalone communications-IC company. Conexant, itself, was a spin-off from the semiconductor division of the well-known and well-regarded Rockwell International Corp. Conexant was known as a leading supplier of innovative semiconductor solutions for imaging, audio, embedded modem, and video surveillance applications.[9] Recently, Conexant was acquired by Synaptics, the leading developer of human interface solutions for over $300 million. Since its formation, Conexant has been an innovator in the semiconductor field (and others) with more than a thousand patents assigned to it.

[9] *See* Conexant's Audio Solution Named CES Innovations 2011 Awards Honoree, BUSINESS WIRE (Nov. 9, 2010), www.businesswire.com/news/home/20101109005618/en/Conexant%E2%80%99s-Audio-Solution-Named-CES-Innovations-2011.

32. The ’959 Patent has been cited by multiple technology companies—as recently as 2017—including, Apple, Fujitsu, IBM, Honeywell, Intel, Matsushita, Oracle, and Samsung.

33. U.S. Patent No. 6,920,573 (“the ’573 Patent”) generally relates to a system for conserving energy in electronic systems. Specifically, the inventor developed a system that provides much-needed energy savings for computers, such as notebooks and laptops, by including various operating modes that limit power usage. In particular, the ’573 Patent describes three operating modes. The first mode is a regular operating mode where the electronic device is fully powered on and where the main microprocessor is running. The second mode is a power-saving mode where the main microprocessor is not running, yet the system is still activated. The third mode is also a power-saving mode, and more specifically, a standby mode from which the first mode can be activated. The ’573 Patent also discloses components to power the system, such as a rechargeable battery, and components to control the system, such as a power button.

34. Major companies in the electronics industry have cited the invention of the ’573 Patent during patent prosecution. Indeed, the ’573 Patent has been cited over fifty times by leading companies, including Compal Electronics, Google, Hewlett-Packard, Intel, Matsushita, Microsoft, NVIDIA, Sony, and Transmeta.

35. U.S. Patent No. 7,493,612 (“the ’612 Patent”) discloses systems and methods for automatically updating the system software of an embedded system. Claim 1 describes an embedded system capable of automatically updating system software using update agent interface programming (UAIP)—code that initiates an update of the system software during the boot process. The embedded system includes first system software and a boot image. The system also includes a micro-controller capable of transforming the first system software into

system code and the boot image into boot code. The boot code includes update agent interface programming (UAIP) for initiating updating of the first system software before executing the system code. The system can be coupled to an external data storage device, which contains the second system software (i.e., the updated system code). If there is an update to the system software, the second system software is read from the external data storage device. As a result of the '612 Patent's inventive system, a computer can advantageously retrieve automatic updates during boot without loading its outdated OS—a more efficient, time-saving solution.

36. The '612 Patent's inventive system was developed by the company, Lite-On Technology Corp., which develops a wide range of consumer electronics products, such as semiconductors, monitors, motherboards, etc. Lite-On was originally founded in 1975 by former employees of Texas Instruments. While the company originally developed LEDs, it branched into other industries, such as embedded systems and related software, and stayed on the forefront of developing technologies. Lite-On was recently purchased by the Japanese company, Kioxia—a former division of Toshiba—for $165 million.

37. The '612 Patent discloses a novel and important invention that is highly relevant to today's technology, which relies heavily on recurring updates to computer systems and IoT devices. It has been cited by major technology companies like Google, IBM, and Texas Instruments.

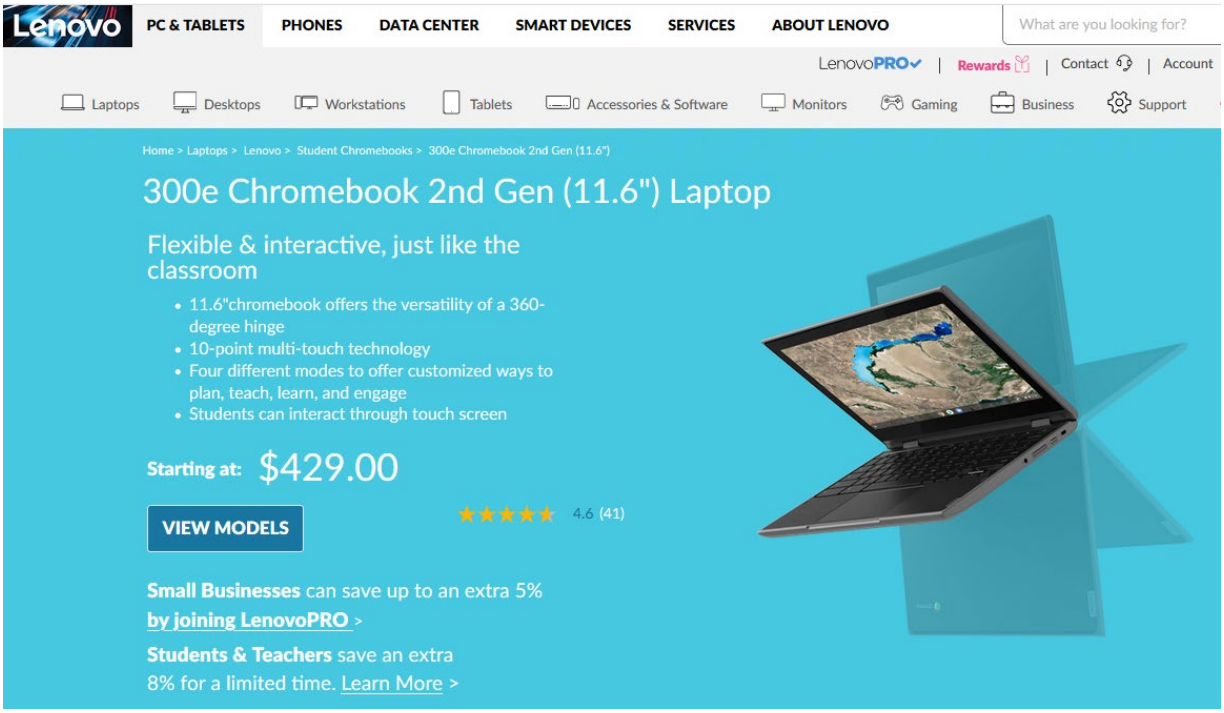
COUNT I

DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,535,959

38. On March 18, 2003, the '959 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Circuit and Method for Reducing Power Consumption in an Instruction Cache."

39. Liberty Patents is the owner of the '959 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '959 Patent against infringers, and to collect damages for all relevant times.

40. Defendant Lenovo made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Lenovo 300e Chromebook 2nd Gen Laptop, and other products that include processors with the capability to ignore reading the tag field when a sequential instruction is to be loaded[10] (processors such as the ARM Cortex-A72, Cortex-A57, Cortex-A15, Cortex-A9, Cortex-R5, Cortex-R4, ARM11, etc.) ("accused products"):



Source: www.lenovo.com/us/en/laptops/lenovo/student-chromebooks/Lenovo-300e-Chromebook-2nd-Gen-MTK/p/88ELC1S9988

[10] *See, e.g.*, Lenovo Chromebook S330, Chromebook C330, Flex 11 Chromebook, Chromebook 100E G2, Chromebook 300E G2 MTK, Chromebook 14" HD Display Business Laptop, 300e Chromebook 2nd Gen (11.6") Laptop, IdeaPad K1, Leez P710.

300e Chromebook (2nd Gen) - Black

Part Number: 81QC0006US

Processor
MediaTek™ 8173C (2.10GHz)

Operating System
Chrome OS

Display Type
11.6" HD (1366 x 768) IPS, anti-glare, touchscreen, 250 nits

Memory
4GB LPDDR3 1866MHz (Soldered)

Hard Drive
32GB eMMC

Source: www.lenovo.com/us/en/laptops/lenovo/student-chromebooks/Lenovo-300e-Chromebook-2nd-Gen-MTK/p/88ELC1S9988

Mediatek MT8173C

The **Mediatek MT8173C** is an upper mainstream ARM SoC (System on a chip) that was introduced in March 2015 and is primarily used for Android based tablets and Chromebooks. It is manufactured in a 28 nm process and has a total of four CPU cores with two Cortex-A72 and two Cortex-A53 cores in a big.LITTLE configuration. The graphics unit is based on the IMG Series 6XT (Rogue) and has the designation PowerVR GX6250. Among others, the GPU includes a video decoder with 4K and H.265 support.

Source: www.notebookcheck.net/Mediatek-MT8173C-Chromebook-SoC.207582.0.html

MT8173

64-bit ARM Cortex-A72/A53 heterogenous multi-processor with CorePilot

MediaTek MT8173 is a highly integrated SOC which incorporates a 64-bit quad-core, with clusters of ARM Cortex-A53 and high performance Cortex-A72 processors operating at up to 2.0GHz. The Imagination PowerVR GX6250 GPU offers OpenGL ES 3.0. To complement, integrated is a high-end 20MP camera ISP, LPDDR3 at up to 933MHz, Ultra HD 2160p video decoding and WQXGA display capability. The MT8173 helps tablet manufacturers to build very high-performance multimedia tablets with a PC-like browser, close to console-level 3D gaming and cinema class home entertainment experiences.

Source: www.mediatek.com/products/tablets/mt8173

41. Defendant Compal made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Lenovo IdeaPad K1 Tablet, and other products that include processors with the capability to ignore reading the tag field when a sequential instruction is to be loaded[11] (processors such as the ARM Cortex-A72, Cortex-A57, Cortex-A15, Cortex-A9, Cortex-R5, Cortex-R4, ARM11, etc.) ("accused products"):

K1 Tablet

Source: https://pcsupport.lenovo.com/us/en/products/tablets/k-series/k1-tablet?linkTrack=Homepage%3ABody_Search%20Products&searchType=3&keyWordSearch=K1%20Tablet

[11] *See, e.g.*, Acer Chromebook 13 CB5-311P; ASUS RT-AC68U Wireless Router; Dell Streak Pro Phone.

Source: https://www.amazon.com/Lenovo-IdeaPad-130422U-10-Inch-Tablet/dp/B0051OKCHG;?language=en_US

Appendix F. Features and Specifications

Model Name: IdeaPad Tablet K1 **Machine Type: 20115/1304**

Note: The following specifications may contain technical inaccuracies or typographical errors. Lenovo reserves the right to improve and/or change specifications at any time without notice.

Form Factor	
Size	264mm x 189mm x 13.3mm
Weight	750g with battery
System	
OS	Android 2.2
CPU	Nvidia Tegra2 T20 1.0GHz

Source: https://fccid.io/GKRPQXU2WB/Users-Manual/Manual-Part-1-1462402 (Page 115)

Nvidia Tegra 2 (250)

The **NVIDIA Tegra 250** is a SOC (System on a Chip) of the Tegra 2 (T20/AP20H) series. It contains a dual-core ARM Cortex-A9 processor, a GeForce ULP graphics card, a DDR2 memory controller, video de- and encoders, a dedicated audio processor and an additional ARM7 core for power saving and management work.

Source: https://www.notebookcheck.net/NVIDIA-Tegra-250-SoC.54654.0.html

42. By doing so, Lenovo and Compal have directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '959 Patent. Defendants' infringement in this regard is ongoing.

43. The ARM Cortex-A72 and the Cortex-A9 are exemplary processors that infringe the '959 Patent.

44. For example, the ARM Cortex-A72 in the Lenovo 300e Chromebook 2nd Gen and the ARM Cortex-A9 in the Lenovo IdeaPad K1 Tablet are processors that each includes an instruction cache. The instruction cache includes multiple cache lines or blocks.

Features

The Cortex-A72 processor includes the following features:

- Full implementation of the ARMv8-A architecture profile. See *1.2 Compliance* on page 1-15.
- Superscalar, variable-length, out-of-order pipeline.
- Dynamic branch prediction with *Branch Target Buffer* (BTB) and *Global History Buffer* (GHB) RAMs, a return stack, and an indirect predictor.
- 48-entry fully-associative L1 instruction *Translation Lookaside Buffer* (TLB) with native support for 4KB, 64KB, and 1MB page sizes.
- 32-entry fully-associative L1 data TLB with native support for 4KB, 64KB, and 1MB page sizes.
- 4-way set-associative unified 1024-entry *Level 2* (L2) TLB in each processor.
- Fixed 48K L1 instruction cache and 32K L1 data cache.
- Shared L2 cache of 512KB, 1MB, 2MB or 4MB configurable size.

Source:

https://static.docs.arm.com/100095/0002/cortex_a72_mpcore_trm_100095_0002_03_en.pdf

(Page 17).

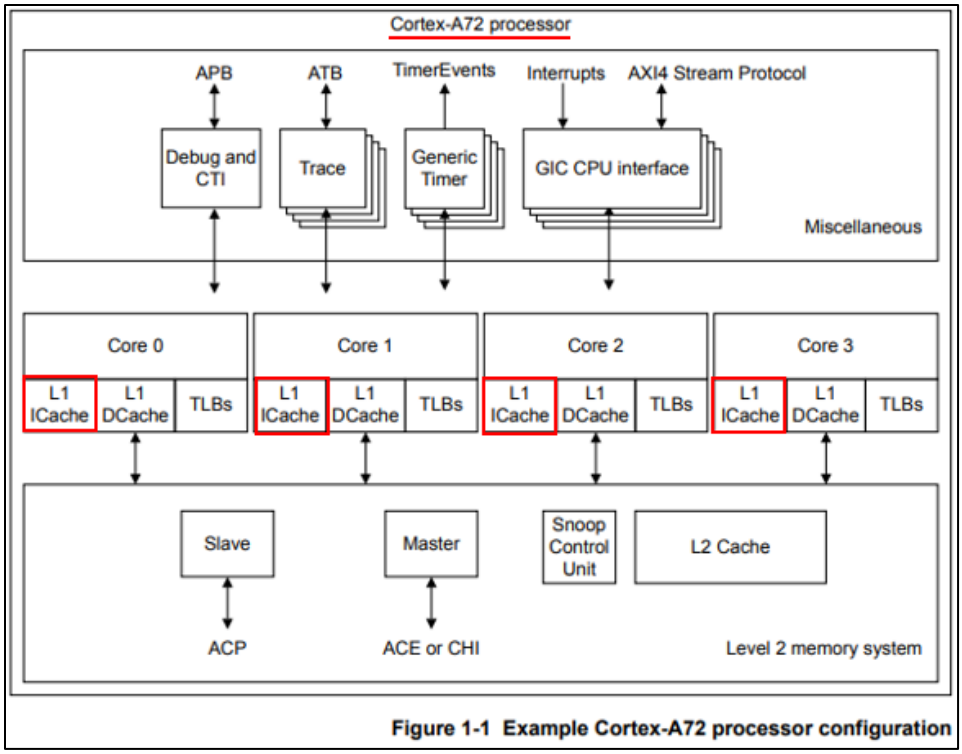


Figure 1-1 Example Cortex-A72 processor configuration

Source:

https://static.docs.arm.com/100095/0002/cortex_a72_mpcore_trm_100095_0002_03_en.pdf

(Page 14).

Instruction fetch

The instruction fetch unit fetches instructions from L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction.

The instruction fetch unit includes:

- L1 instruction cache that is a 48KB 3-way set-associative cache with a 64-byte cache line and optional dual-bit parity protection per 32 bits in the Data RAM and 36 bits in the Tag RAM.
- 48-entry fully-associative L1 instruction *Translation Lookaside Buffer* (TLB) with native support for 4KB, 64KB, and 1MB page sizes.
- 2-level dynamic predictor with *Branch Target Buffer* (BTB) for fast target generation.
- Static branch predictor.
- Indirect predictor.
- Return stack.

Source:

https://static.docs.arm.com/100095/0002/cortex_a72_mpcore_trm_100095_0002_03_en.pdf

(Page 26).

1.1 About the Cortex-A9 processor

The Cortex-A9 processor is a high-performance, low-power, ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A9 processor implements the ARMv7-A architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java bytecodes in Jazelle state.

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 13)

1.6 Configurable options

Table 1-1 shows the configurable options for the Cortex-A9 processor.

Table 1-1 Configurable options for the Cortex-A9 processor

Feature	Range of options
Instruction cache size	16KB, 32KB, or 64KB
Data cache size	16KB, 32KB, or 64KB
TLB entries	64, 128, 256 or 512 entries
BTAC entries	512, 1024, 2048 or 4096 entries

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 19)

11.1 Cache terminology

In a von Neumann architecture, a single cache is used for instruction and data (a unified cache). A modified Harvard architecture has separate instruction and data buses and therefore there are two caches, an instruction cache (I-cache) and a data cache (D-cache). In the ARMv8 processors, there are distinct instruction and data L1 caches backed by a unified L2 cache.

The cache is required to hold an address, some data and some status information.

The following is a brief summary of some of the terms used and a diagram illustrating the fundamental structure of a cache:

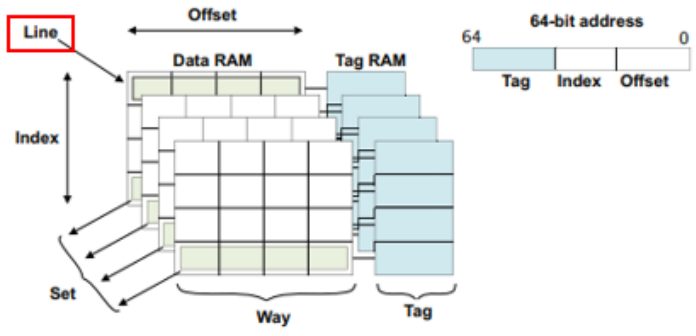


Figure 11-2 Cache terminology

- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

Source:

https://static.docs.arm.com/den0024/a/DEN0024A_v8_architecture_PG.pdf?_ga=2.17157625.1756166971.1588761056-4692096.1569325365 (Page 145).

- It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*, and refers to the smallest loadable unit of a cache, a block of contiguous words from main memory. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

Source:

https://static.docs.arm.com/den0024/a/DEN0024A_v8_architecture_PG.pdf?_ga=2.17157625.1756166971.1588761056-4692096.1569325365 (Page 145).

Cache set	A cache set is a group of cache lines (or blocks). A set contains all the ways that can be addressed with the same index. The number of cache sets is always a power of two. *See also* Cache terminology diagram on the last page of this glossary.
Cache way	A group of cache lines (or blocks). It is 2 to the power of the number of index bits in size. *See also* Cache terminology diagram on the last page of this glossary.

Source:

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0301h/DDI0301H_arm1176jzfs_r0p7_trm.pdf (Page 746).

45. The ARM Cortex-A72 processor in the Lenovo 300e Chromebook 2nd Gen and the ARM Cortex-A9 in the Lenovo IdeaPad K1 Tablet each includes a circuit that is configured to generate a power reduction signal. The power reduction signal indicates if a subsequent instruction to be fetched is in a same block (of a plurality of blocks) as a previous instruction fetched from the instruction cache

46. For example, the ARM Cortex-A72 in the Lenovo 300e Chromebook 2nd Gen and the ARM Cortex-A9 in the Lenovo IdeaPad K1 Tablet each supports a power reduction

method that is operational when an instruction is being accessed from the instruction cache. The instruction cache includes multiple cache lines or blocks, and each cache line or block is associated with a tag value. These tag values are stored in the tag RAM. The cache also includes data RAM for storing the instructions.

47. If a sequential (or subsequent) instruction to be read from the instruction cache is in the same cache line or block as the previous instruction, only the data RAM of the cache is accessed for the instruction, and the tag RAM is *not* accessed because the sequential instruction resides in the same cache line or block.

48. Accordingly, both the ARM Cortex-A72 and Cortex-A9 include a circuit that sends a signal ("power reduction signal") if a sequential instruction to be accessed from the instruction cache is identified as being in the same cache line or block.

L1 instruction memory system

The instruction cache can source up to 128 bits per fetch depending on alignment.

Sequential cache read operations reduce the number of full cache reads. This has the benefit of reducing power consumption. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM way that was previously read is accessed.

Source:

https://static.docs.arm.com/100095/0002/cortex_a72_mpcore_trm_100095_0002_03_en.pdf

(Page 287).

Cache features

The Cortex-A9 processor has separate instruction and data caches. The caches have the following features:

- Each cache can be disabled independently. See *System Control Register* on page 4-25.
- Both caches are 4-way set-associative.
- The cache line length is eight words.
- On a cache miss, critical word first filling of the cache is performed.
- You can configure the instruction and data caches independently during implementation to sizes of 16KB, 32KB, or 64KB.
- To reduce power consumption, the number of full cache reads is reduced by taking advantage of the sequential nature of many cache operations. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM set that was previously read is accessed.

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 113)

11.1 Cache terminology

In a von Neumann architecture, a single cache is used for instruction and data (a unified cache). A modified Harvard architecture has separate instruction and data buses and therefore there are two caches, an instruction cache (I-cache) and a data cache (D-cache). In the ARMv8 processors, there are distinct instruction and data L1 caches backed by a unified L2 cache.

The cache is required to hold an address, some data and some status information.

The following is a brief summary of some of the terms used and a diagram illustrating the fundamental structure of a cache:

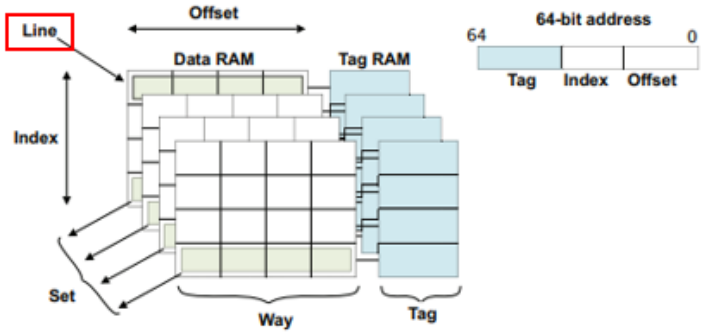


Figure 11-2 Cache terminology

- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

Source:

https://static.docs.arm.com/den0024/a/DEN0024A_v8_architecture_PG.pdf?_ga=2.17157625.1756166971.1588761056-4692096.1569325365 (Page 145).

Instruction fetch

The instruction fetch unit fetches instructions from L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction.

The instruction fetch unit includes:

- L1 instruction cache that is a 48KB 3-way set-associative cache with a 64-byte cache line and optional dual-bit parity protection per 32 bits in the Data RAM and 36 bits in the Tag RAM.
- 48-entry fully-associative L1 instruction *Translation Lookaside Buffer* (TLB) with native support for 4KB, 64KB, and 1MB page sizes.
- 2-level dynamic predictor with *Branch Target Buffer* (BTB) for fast target generation.
- Static branch predictor.
- Indirect predictor.
- Return stack.

Source:

https://static.docs.arm.com/100095/0002/cortex_a72_mpcore_trm_100095_0002_03_en.pdf (Page 26).

- It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*, and refers to the smallest loadable unit of a cache, a block of contiguous words from main memory. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

Source:

https://static.docs.arm.com/den0024/a/DEN0024A_v8_architecture_PG.pdf?_ga=2.17157625.1756166971.1588761056-4692096.1569325365 (Page 145).

49. Defendants Lenovo and Compal have had knowledge of the '959 Patent at least as of the date when they were notified of the filing of this action.

50. Liberty Patents has been damaged as a result of the infringing conduct by Defendants alleged above. Thus, Lenovo and Compal are liable to Liberty Patents in an amount

that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

51. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the '959 Patent.

COUNT II

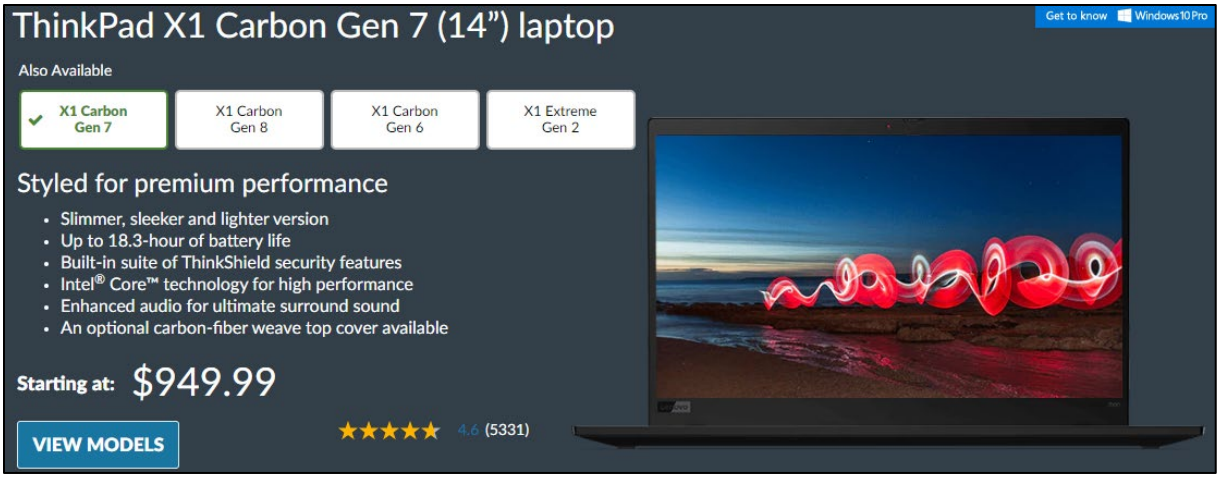
DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,920,573

52. On July 19, 2005, the '573 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Energy-Conserving Apparatus and Operating System Having Multiple Operating Functions Stored in Keep-Alive Memory."

53. Liberty Patents is the owner of the '573 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '573 Patent against infringers, and to collect damages for all relevant times.

54. Defendant Lenovo made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Lenovo ThinkPad X1 Carbon Gen 7 laptop and other products including the "Always On USB" feature[12] ("accused products"):

[12] *See, e.g.*, Lenovo G570, IdeaPad Flex (14, 15), IdeaPad L340, IdeaPad S145, IdeaPad S940, Legion Y540, Legion Y7000, ThinkPad 10 Tablet, ThinkPad E580, ThinkPad E595, ThinkPadP43, ThinkPadP53, ThinkPad T400, ThinkPad T440, ThinkPad T490, ThinkPad T495, ThinkPad T590, ThinkPad W530, ThinkPad X1 Carbon, ThinkPad X1 Yoga, ThinkPad Yoga 260, Yoga C630 (13Q50), Yoga 900S, Yoga C740, Yoga C940, V130-15IKB.



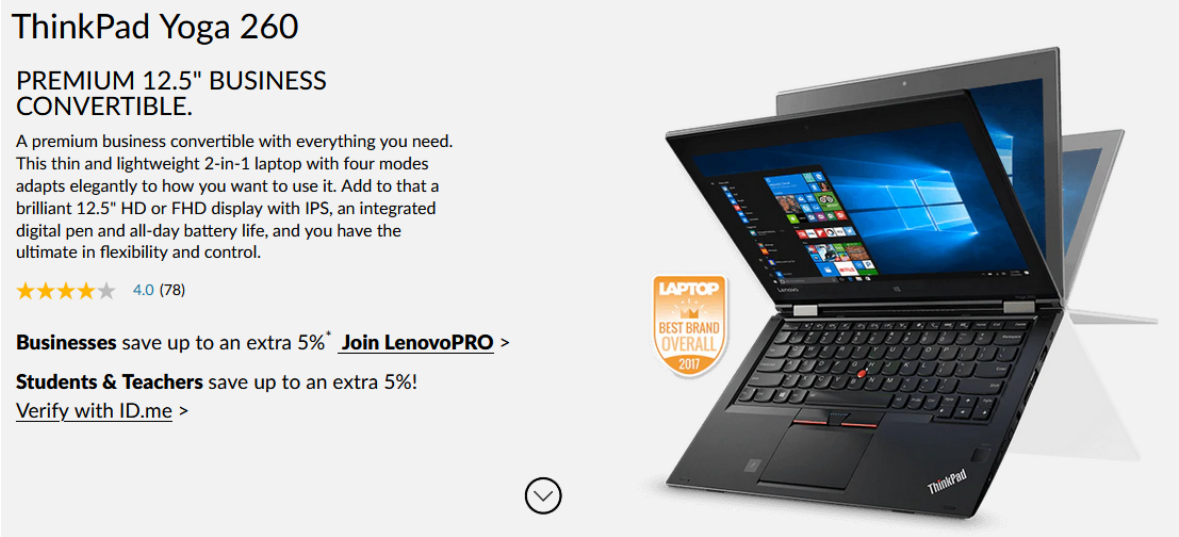
Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x1/X1-Carbon-Gen-7/p/22TP2TXX17G

3. Always On USB 3.1 connector Gen 1	With the Always On USB feature enabled, the Always On USB 3.1 connector Gen 1 can charge a USB-compatible device when the computer is on, off, in sleep mode, or in hibernation mode. **Notes:** • By default, **Always On USB** is enabled and **Charge in Battery Mode** is disabled in UEFI BIOS. • When the computer is off or in hibernation mode, and **Charge in Battery Mode** is disabled in UEFI BIOS, ensure that you have connected the computer to ac power. To enable the Always On USB feature: 1. Enter the UEFI BIOS menu. See "Enter the UEFI BIOS menu" on page 47. 2. Click **Config → USB → Always On USB** to enable the Always On USB feature.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf (Page 7)

55. Defendant Compal made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Lenovo ThinkPad Yoga 260 notebook and other products with the capability to charge devices while the main microprocessor is not running[13] ("accused products"):

[13] *See, e.g.*, Acer Nitro 5 (includes "Power-Off Charging" feature); Dell XPS 13 UltraBook (includes "PowerShare" feature); HP Spectre 13 (includes "Sleep and Charge" feature).



Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-yoga/Yoga-260/p/22TP2TXY260

Left-side view

1 Power connector	2 OneLink+ connector
3 Mini DisplayPort connector	4 Always On USB connector (USB 3.0 connector
5 Smart card slot (available on some models)	

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984; https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 5).

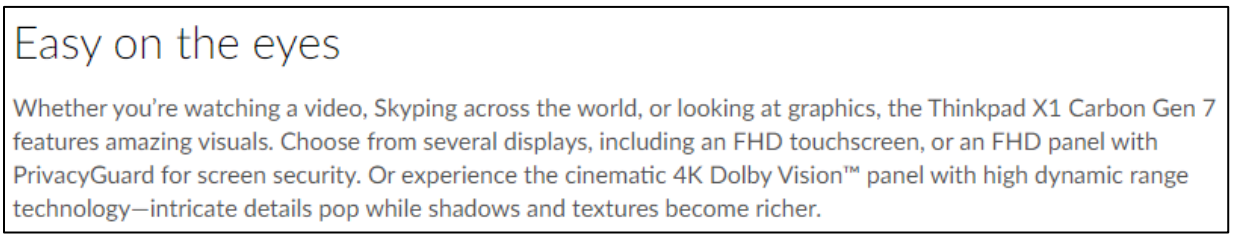
56. By doing so, Defendants have directly infringed (literally and/or under the doctrine of equivalents) at least Claim 13 of the '573 Patent. Defendants' infringement of the '573 Patent is ongoing.

57. The Lenovo ThinkPad X1 Carbon and ThinkPad Yoga 260 are exemplary products that infringe the '573 Patent.

58. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 are information-processing apparatuses with multiple operating functions. Each includes a first group of circuitry that is actuatable to provide a first operating function. The first group of circuitry comprises main microprocessor circuitry.

59. For example, the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each include a processor for performing various processing functions. The processor includes Arithmetic Logic Units (ALU), Instruction and Data Caches, and other blocks. The processor also has different states like working state, sleeping state, and off state etc., which correspond to the laptop's Power On mode, Sleep mode and Shut Down mode, respectively. The processor functions differently depending on the current operating mode.

60. During Power On mode, the processor provides processing functions, including application processing, graphics processing, etc. ("first operating function"). The processing blocks like ALU, FPU, memory etc. ("first group of circuitry") consume power and implement these required functions. These blocks are part of the core or Central Processing Unit ("main microprocessor circuitry") of the processor.



Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x1/X1-Carbon-Gen-7/p/22TP2TXX17G

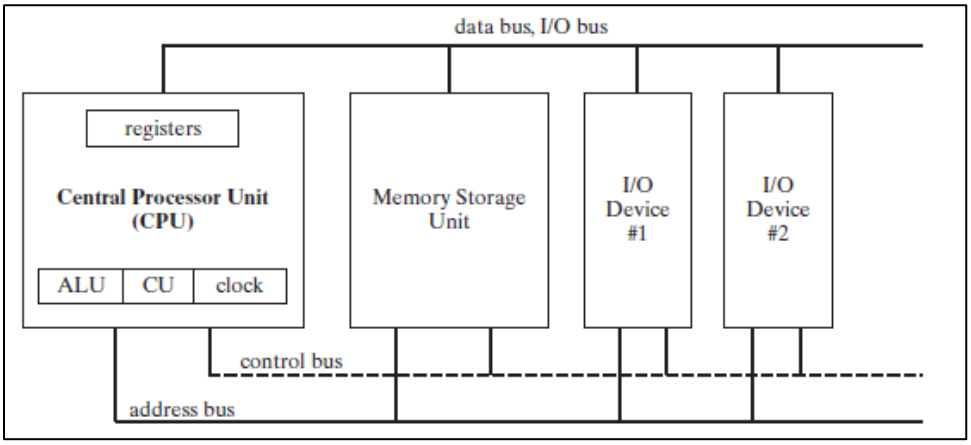
Performance Plus

PC performance has never been faster. Intel® 6th Gen Core™ i processors with built-in security are designed to take your productivity, creativity, and 3D gaming to the next level. And with Windows 10 you can empower yourself and your business to do great things across many devices securely anywhere, with anyone, and at any time. With up to 16 GB of memory and up to 512 GB of SSD storage, The ThinkPad Yoga 260 can keep up with you.

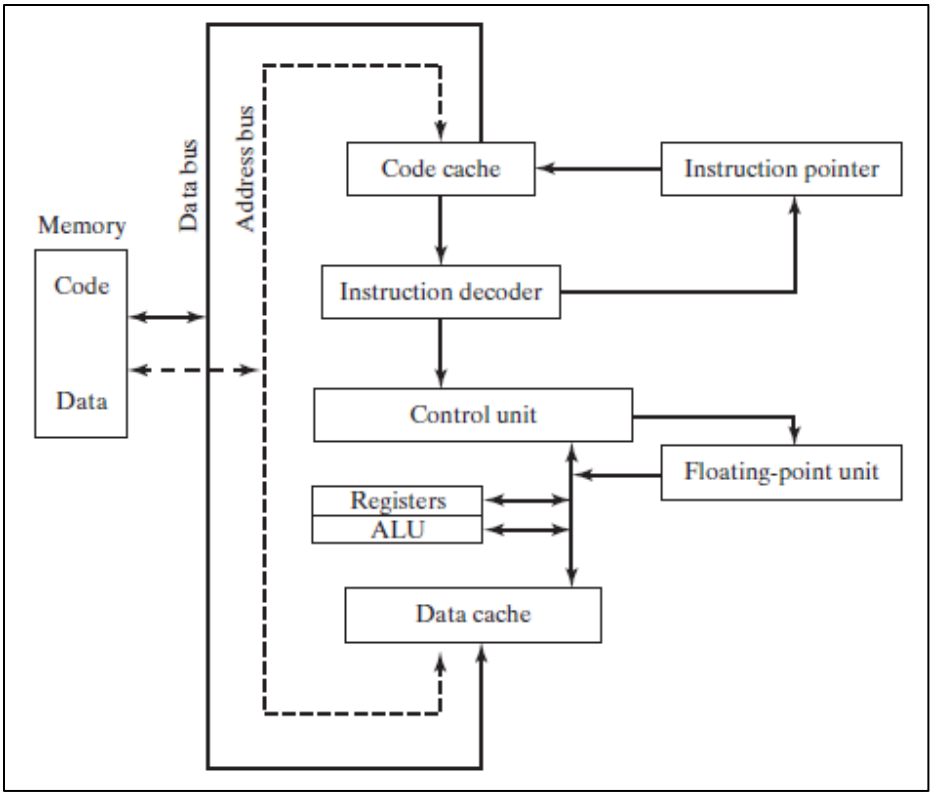
Four Awesome Modes. One Incredible Device.

Easily change between four modes — Laptop, Stand, Tablet, and Tent — while your system automatically switches system settings. The secret: ThinkPad Yoga 260's unique design, improved hinge, and touchscreen display allow the screen to flip around 360 degrees – and even lay flat – which means you can select the mode you use based on what you choose to do.

Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-yoga/Yoga-260/p/22TP2TXY260



Source: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-x86-processor-architecture/



Source: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-x86-processor-architecture/

61. The following citations disclose different operating modes of the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260, including Shut Down mode, Sleep mode, and Power On mode. The computers operate differently according to the current operating mode.

ThinkPad X1 Carbon Gen 7

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

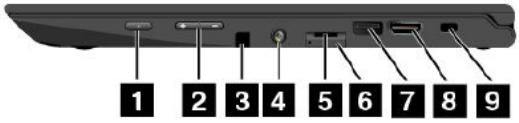
(Page 6)

2. Power button	Press to turn on the computer or put the computer to sleep mode. To turn off the computer, open the Start menu, click **Power**, and then select **Shut down**. The indicator in the power button shows the system status of your computer. • **Blinking for three times:** The computer is initially connected to power. • **On:** The computer is on. • **Off:** The computer is off or in hibernation mode. • **Blinking rapidly:** The computer is entering sleep or hibernation mode. • **Blinking slowly:** The computer is in sleep mode.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 6)

ThinkPad Yoga 260



1 Power button	2 Volume-control buttons
3 ThinkPad Pen Pro (available on some models)	4 Audio connector
5 Micro-SD-card slot	6 Micro-SIM-card tray (available on some models)
7 USB 3.0 connector	8 HDMI™ connector
9 Security-lock slot	

1 Power button

Press the power button to turn on the computer or put the computer into sleep mode. For information about how to turn off the computer, see "Frequently asked questions" on page 15.

If your computer is unresponsive, you can turn off the computer by pressing and holding the power button for four or more seconds. If the computer cannot be turned off, see "Computer stops responding" on page 99 for more information.

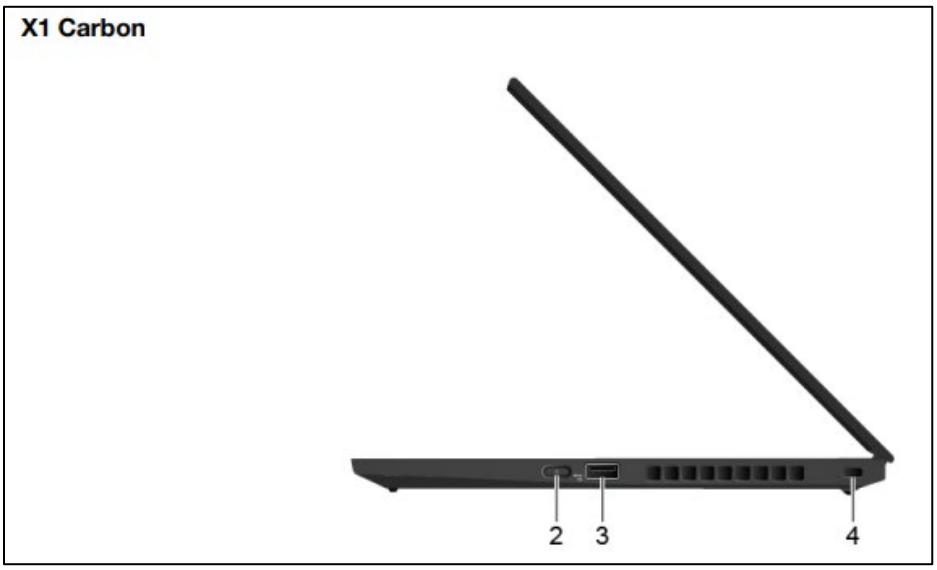
Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984; https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 2).

62. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes a second group of circuitry that is actuatable to provide a second operating function. During the second operating function, the system is not required to activate the main microprocessor circuitry.

63. For example, both the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 include the "Always On USB" feature that allows a user to charge ("second operating function") USB connected devices (such as such as a mobile devices, cameras, activity trackers, smartwatches, etc.) even when the laptop is in the Shut down or Off mode. Mobile devices can be charged using the designated USB port having the "Always on USB" feature without requiring the laptop to be in working state (i.e., Power On mode). The corresponding USB

charger IC/USB board circuit ("second group of circuitry") can be actuated to provide the charging function during Shut Down mode.

ThinkPad X1 Carbon Gen 7



Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf (Page 6)

Ports	• 2 x USB 3.1 (Gen 2) Type-C / Intel Thunderbolt™ 3 (DisplayPort, Data transfer) • 2 x USB 3.1 (Gen 1)** (1 always on) • HDMI 1.4 • Network extension for Ethernet/side mechanical docking • Headphone / mic combo • Optional: WWAN slot

Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x1/X1-Carbon-Gen-7/p/22TP2TXX17G

3. Always On USB 3.1 connector Gen 1	With the Always On USB feature enabled, the Always On USB 3.1 connector Gen 1 can charge a USB-compatible device when the computer is on, off, in sleep mode, or in hibernation mode. **Notes:** • By default, **Always On USB** is enabled and **Charge in Battery Mode** is disabled in UEFI BIOS. • When the computer is off or in hibernation mode, and **Charge in Battery Mode** is disabled in UEFI BIOS, ensure that you have connected the computer to ac power. To enable the Always On USB feature: 1. Enter the UEFI BIOS menu. See "Enter the UEFI BIOS menu" on page 47. 2. Click **Config → USB → Always On USB** to enable the Always On USB feature.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 7)

ThinkPad Yoga 260

1 Power connector	2 OneLink+ connector
3 Mini DisplayPort connector	4 Always On USB connector (USB 3.0 connector
5 Smart card slot (available on some models)	

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984;

https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 4).

4 Always On USB connector (USB 3.0 connector)

By default, the Always On USB connector enables you to charge some mobile digital devices and smartphones when your computer is in the following situation:

- When your computer is on or in sleep mode
- When your computer is in hibernation mode or is turned off, but connected to ac power

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984;

https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 5).

64. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes a third group of circuitry that is actuatable to provide a standby function that allows the first group

of circuitry (when deactivated) to be reactuatable so that it can provide the first operating function. The third group of circuitry also comprises keep-alive memory circuitry for storing information needed for resuming the first operating function or the second operating function.

65. For example, both the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 include different operating modes like Sleep mode, Power On mode, and Shut down mode. The Sleep mode ("standby function") can be activated and deactivated (i.e., to wake up the system) by pressing the Power button. The laptop includes corresponding circuitry ("third group of circuitry") that activates and deactivates the Sleep mode.

66. During Sleep mode, computational tasks are not performed, and the system consumes less power. The system retains enough context in order to return to a working state ("resuming said first operating function") by storing or saving information in hardware memory, such as RAM or in a disk ("keep-alive memory circuitry").

ThinkPad X1 Carbon Gen 7

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf (Page 6)

2. Power button	Press to turn on the computer or put the computer to sleep mode. To turn off the computer, open the Start menu, click **Power**, and then select **Shut down**. The indicator in the power button shows the system status of your computer. • **Blinking for three times:** The computer is initially connected to power. • **On:** The computer is on. • **Off:** The computer is off or in hibernation mode. • **Blinking rapidly:** The computer is entering sleep or hibernation mode. • **Blinking slowly:** The computer is in sleep mode.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

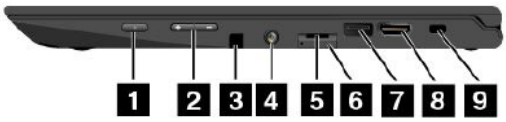
(Page 6)

Fn+4	Put the computer to sleep mode. To wake up the computer, press Fn or the power button.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 19)

ThinkPad Yoga 260



1 Power button	2 Volume-control buttons
3 ThinkPad Pen Pro (available on some models)	4 Audio connector
5 Micro-SD-card slot	6 Micro-SIM-card tray (available on some models)
7 USB 3.0 connector	8 HDMI™ connector
9 Security-lock slot	

1 Power button

Press the power button to turn on the computer or put the computer into sleep mode. For information about how to turn off the computer, see "Frequently asked questions" on page 15.

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984;

https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 2, page 17).

67. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes power providing means for providing power to the first group of circuitry, the second group of circuitry, and the third group of circuitry.

68. For example, both the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 include a battery ("power providing means") for providing power to the different circuits present in the system, including the CPU, memory, and I/O Peripherals (which include USB).

ThinkPad X1 Carbon Gen 7

Battery	Up to 18.3 hours* on 51Wh, includes Rapid Charge technology† *All battery life claims are approximate and based on results using both the MobileMark® 2014 battery-life benchmark tests. Actual battery life will vary and depends on many factors such as product configuration and usage, software use, wireless functionality, power management settings, and screen brightness. The maximum capacity of the battery will decrease with time and use. †Requires 65W AC power adapter

Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x1/X1-Carbon-Gen-7/p/22TP2TXX17G

What's in the box	• ThinkPad X1 Carbon Gen 7 • 65W AC adapter • 4 cell 80Wh or 4 cell 51Wh internal battery • Quick start guide

Source: https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x1/X1-Carbon-Gen-7/p/22TP2TXX17G

ThinkPad Yoga 260

Power management

The power to run your computer can come either from ac power or from the battery.

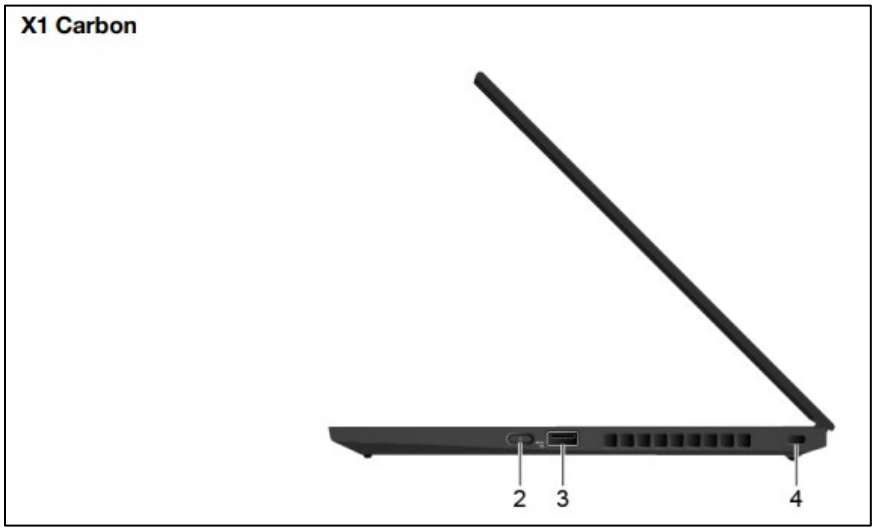
Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984; https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 34).

69. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes control means for controlling said power providing means to selectively activate said first group of circuitry, said second group of circuitry, and said third group of circuitry, so as to respectively provide said first operating function, said second operating function, and said standby function.

70. The Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes different operating modes like Power On, Sleep, and Shut Down modes. Sleep mode ("standby function"), Shut Down mode, and Power On mode (which provides "first operating function") can be activated using the Power button ("control means"). The USB port with the "Always On USB" feature enables charging of a mobile device through the designated USB port during Shut Down mode ("second operating function").

71. The processors of the Lenovo ThinkPad X1 Carbon Gen 7 and ThinkPad Yoga 260 each includes a Power Management Integrated Circuit (PMIC) that manages the power distribution in the processor system. The PMIC provides power to different circuits of the processor system. Further, the PMIC receives control inputs from the processor system, i.e., signals from the power button are used by PMIC as control inputs for enabling and disabling the power distribution for the circuits in the processor system.

ThinkPad X1 Carbon Gen 7



Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 6)

2. Power button	Press to turn on the computer or put the computer to sleep mode. To turn off the computer, open the Start menu, click **Power**, and then select **Shut down**. The indicator in the power button shows the system status of your computer. • **Blinking for three times:** The computer is initially connected to power. • **On:** The computer is on. • **Off:** The computer is off or in hibernation mode. • **Blinking rapidly:** The computer is entering sleep or hibernation mode. • **Blinking slowly:** The computer is in sleep mode.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 6)

Set power button behaviors

You can define what the power button does according to your preference. For example, by pressing the power button, you can turn off the computer or put the computer to sleep or hibernation mode.

To change what the power button does:

1. Right-click the battery status icon in the Windows notification area.
2. Click **Power Options → Choose what the power buttons do**.
3. Change the settings as you prefer.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 32)

What do I do if my computer stops responding?	1. Press and hold the power button until the computer turns off. Then, restart the computer. 2. If step 1 does not work: • For models with an emergency reset hole: Insert a straightened paper clip into the emergency reset hole to cut off power supply temporarily. Then, restart the computer with ac power connected. • For models without an emergency reset hole: – For models with the removable battery, remove the removable battery and disconnect all power sources. Then, reconnect to ac power and restart the computer. – For models with the built-in battery, disconnect all power sources. Press and hold the power button for about seven seconds. Then, reconnect to ac power and restart the computer.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

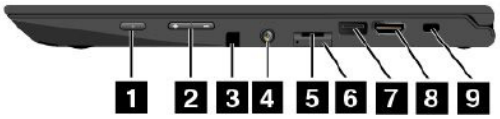
(Page 67)

3. Always On USB 3.1 connector Gen 1	With the Always On USB feature enabled, the Always On USB 3.1 connector Gen 1 can charge a USB-compatible device when the computer is on, off, in sleep mode, or in hibernation mode. **Notes:** • By default, **Always On USB** is enabled and **Charge in Battery Mode** is disabled in UEFI BIOS. • When the computer is off or in hibernation mode, and **Charge in Battery Mode** is disabled in UEFI BIOS, ensure that you have connected the computer to ac power. To enable the Always On USB feature: 1. Enter the UEFI BIOS menu. See "Enter the UEFI BIOS menu" on page 47. 2. Click **Config → USB → Always On USB** to enable the Always On USB feature.

Source: https://download.lenovo.com/pccbbs/mobiles_pdf/x1yoga_x1carbon_ug_v2_en.pdf

(Page 7)

ThinkPad Yoga 260



1 Power button	2 Volume-control buttons
3 ThinkPad Pen Pro (available on some models)	4 Audio connector
5 Micro-SD-card slot	6 Micro-SIM-card tray (available on some models)
7 USB 3.0 connector	8 HDMI™ connector
9 Security-lock slot	

1 Power button

Press the power button to turn on the computer or put the computer into sleep mode. For information about how to turn off the computer, see "Frequently asked questions" on page 15.

If your computer is unresponsive, you can turn off the computer by pressing and holding the power button for four or more seconds. If the computer cannot be turned off, see "Computer stops responding" on page 99 for more information.

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984;

https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 2).

You also can define what the power button does. For example, by pressing the power button, you can turn off the computer or put the computer into sleep or hibernation mode. To change what the power button does, do the following:

1. Go to Control Panel and then change the view of Control Panel from Category to Large icons or Small icons.
2. Click **Power Options → Choose what the power buttons do.**
3. Follow the instructions on the screen.

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984;

https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 3).

1 Power connector	2 OneLink+ connector
3 Mini DisplayPort connector	4 Always On USB connector (USB 3.0 connector
5 Smart card slot (available on some models)	

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984; https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 4).

4 Always On USB connector (USB 3.0 connector)

By default, the Always On USB connector enables you to charge some mobile digital devices and smartphones when your computer is in the following situation:

- When your computer is on or in sleep mode
- When your computer is in hibernation mode or is turned off, but connected to ac power

Source: https://fccid.io/GKR-TP00078ASI/Users-Manual/Users-Manual-2933984; https://download.lenovo.com/pccbbs/mobiles_pdf/yoga260_ug_en.pdf (page 5).

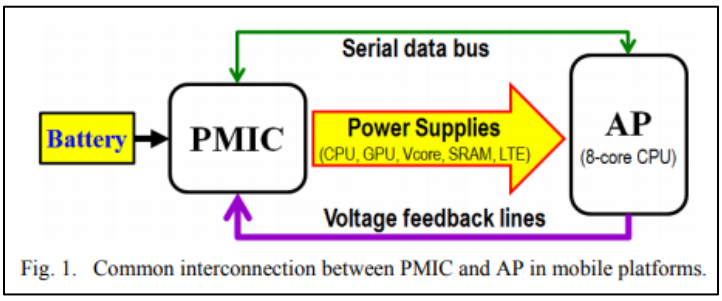


Fig. 1. Common interconnection between PMIC and AP in mobile platforms.

Source: https://ieeexplore.ieee.org/document/7237388

72. Defendants Lenovo and Compal have had knowledge of the '573 Patent at least as of the date when they were notified of the filing of this action.

73. Liberty Patents has been damaged as a result of Defendants' infringing conduct alleged above. Thus, Defendants are liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

74. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the '573 Patent.

COUNT III

DIRECT INFRINGEMENT OF U.S. PATENT NO. 7,493,612

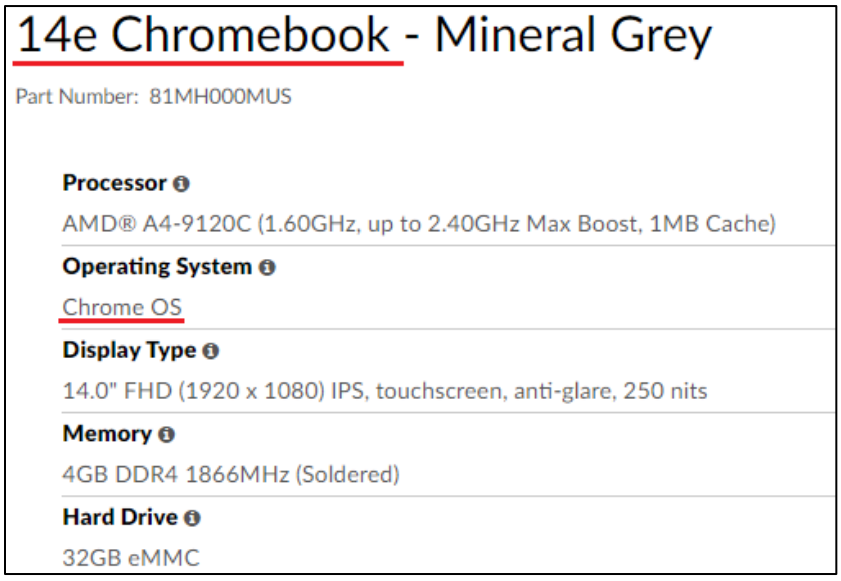
75. On February 17, 2009, the '612 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Embedded System and Related Method Capable of Automatically Updating System Software."

76. Liberty Patents is the owner of the '612 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '612 Patent against infringers, and to collect damages for all relevant times.

77. Defendant Lenovo made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Lenovo 14e Chromebook, which is part of the Chromebook family of products running Chrome OS[14] ("accused products"):

[14] *See, e.g.*, Lenovo 10e Chromebook Tablet, 100S Chromebook, Chromebook C340-15, Chromebook C340-11, Chromebook S340-14, Chromebook S345-14, Flex 11 Chromebook, N20 Chromebook, N21 Chromebook, N22 Chromebook, N23 Chromebook, 100e Chromebook, 100e Chromebook 2nd Gen, 100e Chromebook 2nd Gen MTK, N23 Yoga Chromebook, 300e Chromebook, 300e Chromebook 2nd Gen, 300e Chromebook 2nd Gen MTK, 500e Chromebook, 500e Chromebook 2nd Gen, N42 Chromebook, 14e Chromebook, IdeaPad S330 Chromebook, IdeaPad C330 Chromebook, Yoga C630 Chromebook, ThinkPad 11e Chromebook, ThinkPad 11e 3rd Gen Chromebook, ThinkPad 11e 4th Gen Chromebook, ThinkPad 13.

Source: https://www.lenovo.com/us/en/laptops/lenovo/student-chromebooks/Lenovo-14e-Chromebook/p/88ELC1S9991



Source: https://www.lenovo.com/us/en/laptops/lenovo/student-chromebooks/Lenovo-14e-Chromebook/p/88ELC1S9991

Auto Update policy

Overview

Chrome devices (e.g. Chromebook, Chromebox, Chromebase, Chromebit) receive automatic updates that enhance both the device and its software. Device updates provide the latest features and keep the device secure, and are applied across the operating system, browser and hardware. These updates depend on many device specific non-Google hardware and software providers that work with Google to provide the highest level of security and stability support. For this reason, older Chrome devices cannot receive updates indefinitely to enable new OS and browser features.

Source: https://support.google.com/chrome/a/answer/6220366?hl=en

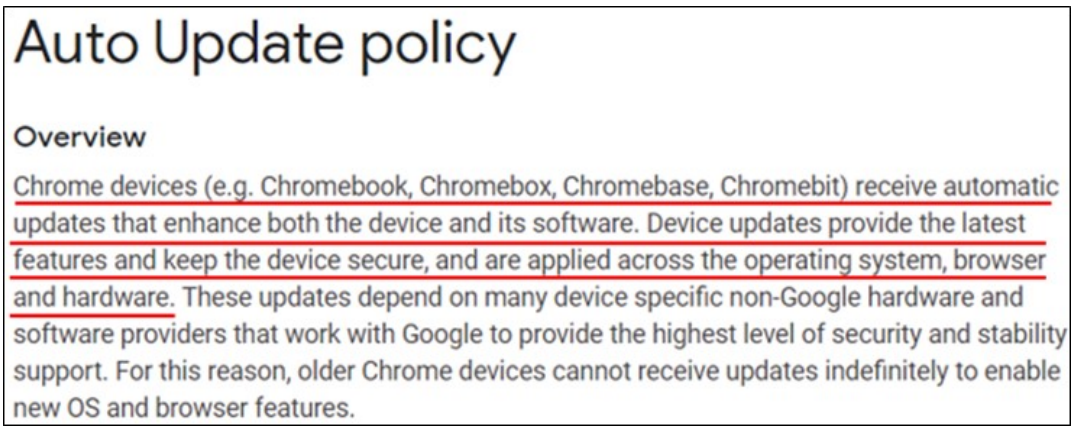
78. Defendant Compal made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, the Acer Chromebook C710, which is part of the Chromebook family of products running Chrome OS[15] ("accused products"):

Source: https://www.pcmag.com/reviews/acer-c7-chromebook-c710-2055?test_uuid=03VgQESfXzdGbpM9u5nO4tq&test_variant=b

[15] *See also, e.g.*, Dell Chromebook 11 3180.

SYSTEM	
Notebook Type	Chromebook
Platform	Chrome OS
Hard Drive Capacity	320 GB
Embedded Security	Trusted Platform Module (TPM) Security Chip

Source: https://www.cnet.com/products/acer-c710-2055-chromebook/specs/



Auto Update policy

Overview

Chrome devices (e.g. Chromebook, Chromebox, Chromebase, Chromebit) receive automatic updates that enhance both the device and its software. Device updates provide the latest features and keep the device secure, and are applied across the operating system, browser and hardware. These updates depend on many device specific non-Google hardware and software providers that work with Google to provide the highest level of security and stability support. For this reason, older Chrome devices cannot receive updates indefinitely to enable new OS and browser features.

Source: https://support.google.com/chrome/a/answer/6220366?hl=en

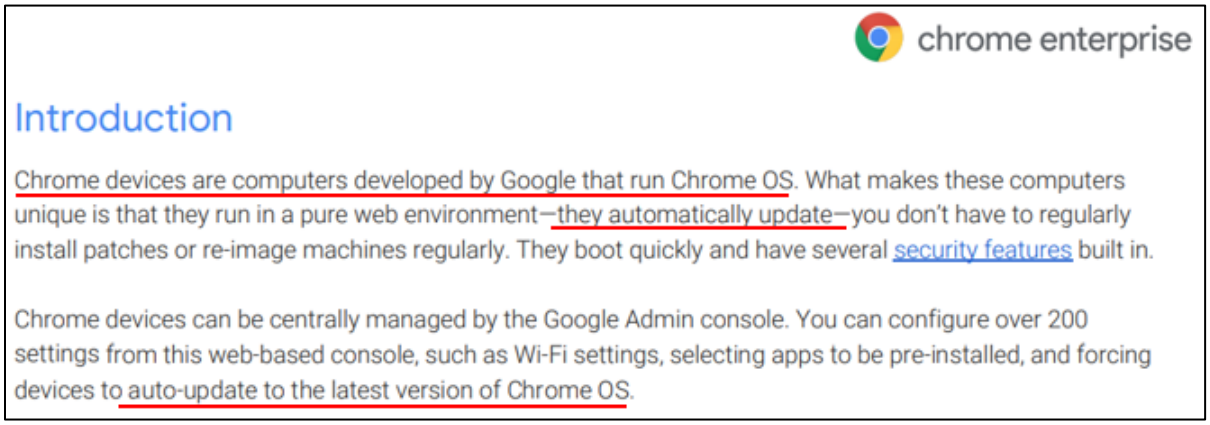
79. By doing so, Defendants have directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '612 Patent. Defendants' infringement in this regard is ongoing.

80. The Lenovo 14e Chromebook and the Acer Chromebook C710 are exemplary products that infringe the '612 Patent

81. The Lenovo 14e Chromebook and the Acer Chromebook C710 each includes an embedded system that is capable of automatically updating system software.

82. For example, the Lenovo 14e Chromebook and the Acer Chromebook C710 both run Chrome OS, which can automatically update the device's firmware over the internet. Chromebooks include an Embedded Controller (EC), which is responsible for power sequencing

of the main CPU or the Application Processor (AP), keyboard control, thermal control, battery charging control, verified boot, etc. The EC—together with the components it controls—is an embedded system capable of automatically updating the system software. Specifically, the system includes two types of firmware: RO firmware and RW firmware. The RW firmware ("system software") is stored in the updateable section of the firmware and can be automatically updated.

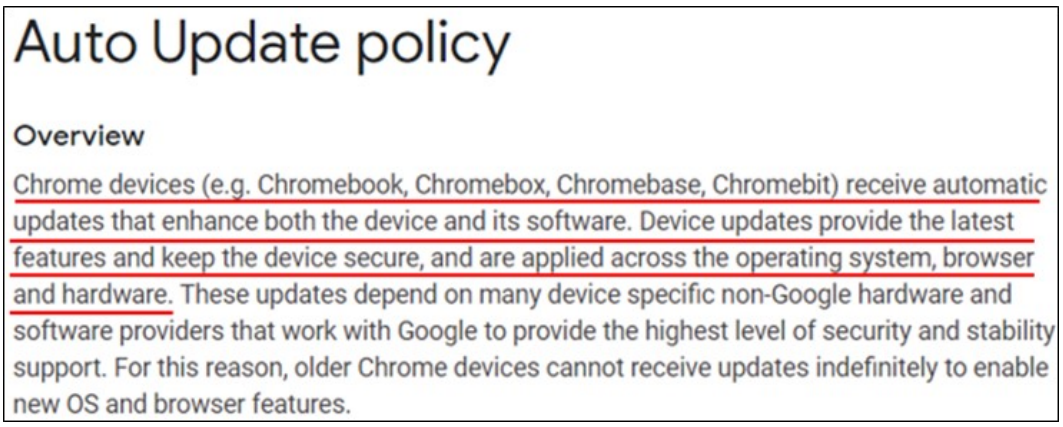

chrome enterprise

Introduction

Chrome devices are computers developed by Google that run Chrome OS. What makes these computers unique is that they run in a pure web environment—they automatically update—you don't have to regularly install patches or re-image machines regularly. They boot quickly and have several security features built in.

Chrome devices can be centrally managed by the Google Admin console. You can configure over 200 settings from this web-based console, such as Wi-Fi settings, selecting apps to be pre-installed, and forcing devices to auto-update to the latest version of Chrome OS.

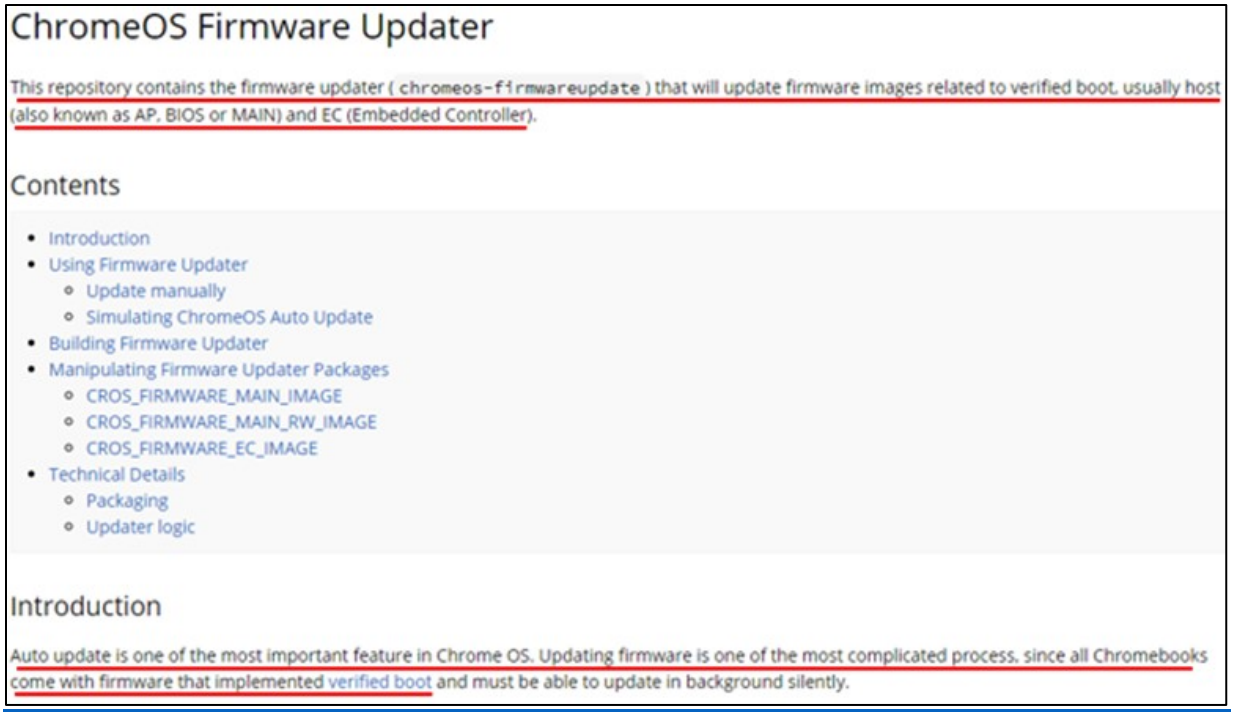
Source: https://services.google.com/fh/files/misc/chrome_device_deployment_guide.pdf (Page 4)


Auto Update policy

Overview

Chrome devices (e.g. Chromebook, Chromebox, Chromebase, Chromebit) receive automatic updates that enhance both the device and its software. Device updates provide the latest features and keep the device secure, and are applied across the operating system, browser and hardware. These updates depend on many device specific non-Google hardware and software providers that work with Google to provide the highest level of security and stability support. For this reason, older Chrome devices cannot receive updates indefinitely to enable new OS and browser features.

Source: https://support.google.com/chrome/a/answer/6220366?hl=en



ChromeOS Firmware Updater

This repository contains the firmware updater (`chromeos-firmwareupdate`) that will update firmware images related to verified boot, usually host (also known as AP, BIOS or MAIN) and EC (Embedded Controller).

Contents

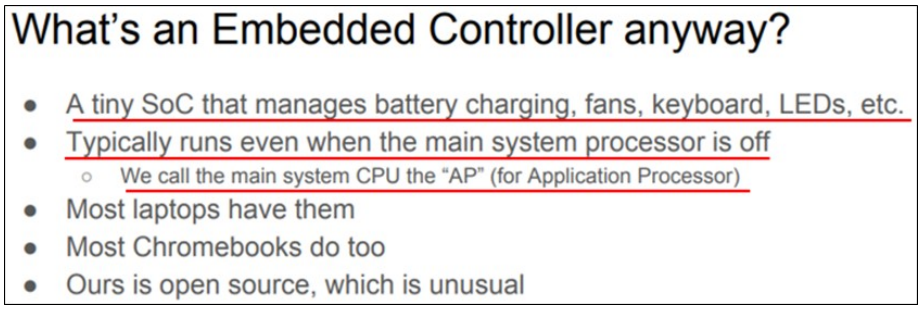
- Introduction
- Using Firmware Updater
  - Update manually
  - Simulating ChromeOS Auto Update
- Building Firmware Updater
- Manipulating Firmware Updater Packages
  - CROS_FIRMWARE_MAIN_IMAGE
  - CROS_FIRMWARE_MAIN_RW_IMAGE
  - CROS_FIRMWARE_EC_IMAGE
- Technical Details
  - Packaging
  - Updater logic

Introduction

Auto update is one of the most important feature in Chrome OS. Updating firmware is one of the most complicated process, since all Chromebooks come with firmware that implemented verified boot and must be able to update in background silently.

Source:

https://chromium.googlesource.com/chromiumos/platform/firmware/+/master/README.md



What's an Embedded Controller anyway?

- A tiny SoC that manages battery charging, fans, keyboard, LEDs, etc.
- Typically runs even when the main system processor is off
  - We call the main system CPU the "AP" (for Application Processor)
- Most laptops have them
- Most Chromebooks do too
- Ours is open source, which is unusual

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 4).

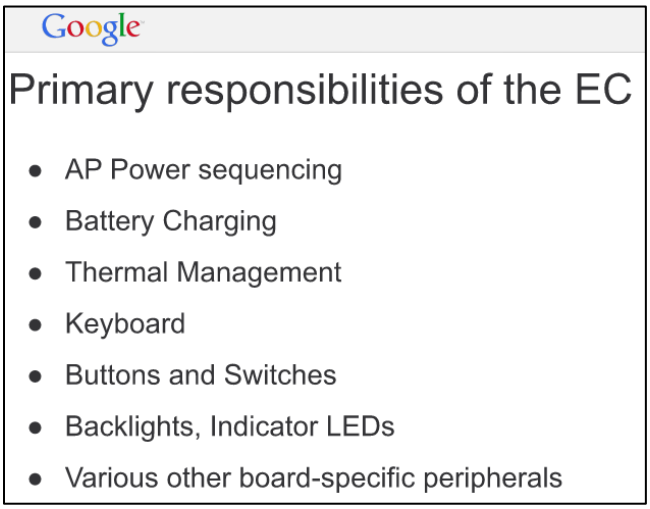
Introduction

The Chromium OS project includes open source software for embedded controllers (EC) used in recent ARM and x86 based Chromebooks. This software includes a lightweight, multitasking OS with modules for power sequencing, keyboard control, thermal control, battery charging, and verified boot. The EC software is written in C and supports a variety of micro-controllers.

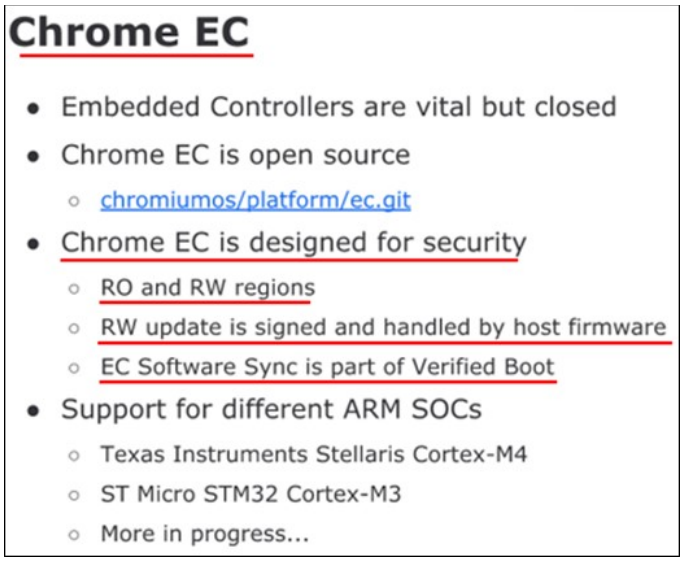
This document is a guide to help make you familiar with the EC code, current features, and the process for submitting code patches.

For more see the Chrome OS Embedded Controller presentation and video from the 2014 Firmware Summit.

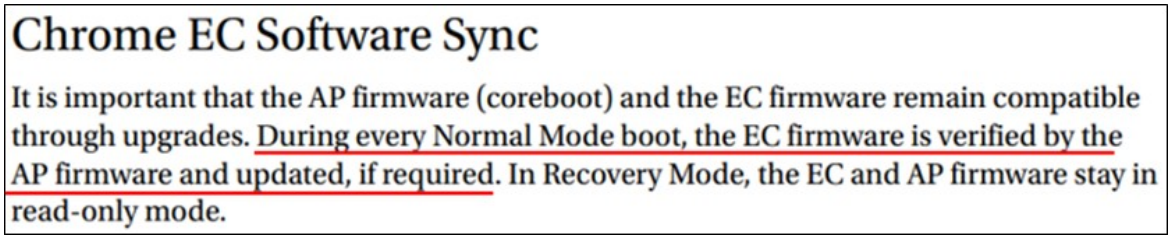
Source: https://chromium.googlesource.com/chromiumos/platform/ec/



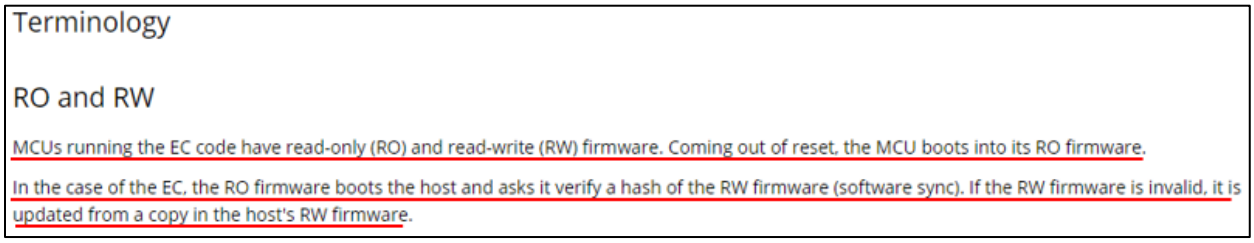
Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8__TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bc16935c_0142 (Slide 20).



Source: https://docs.google.com/presentation/d/1h-nsDGlQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298 (Slide 29).

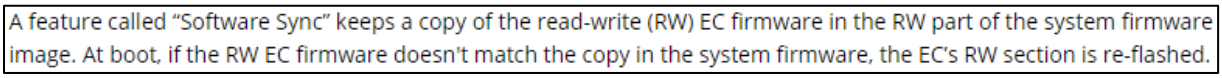


Source: https://link.springer.com/content/pdf/10.1007%2F978-1-4842-0070-4.pdf (Page 119).



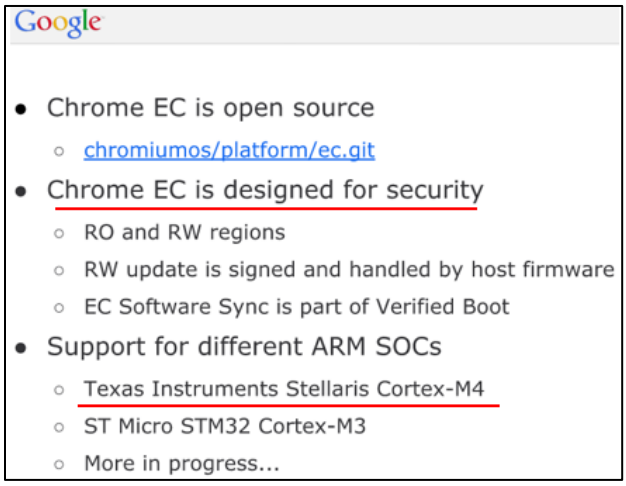
Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+/master/docs/write_protection.md

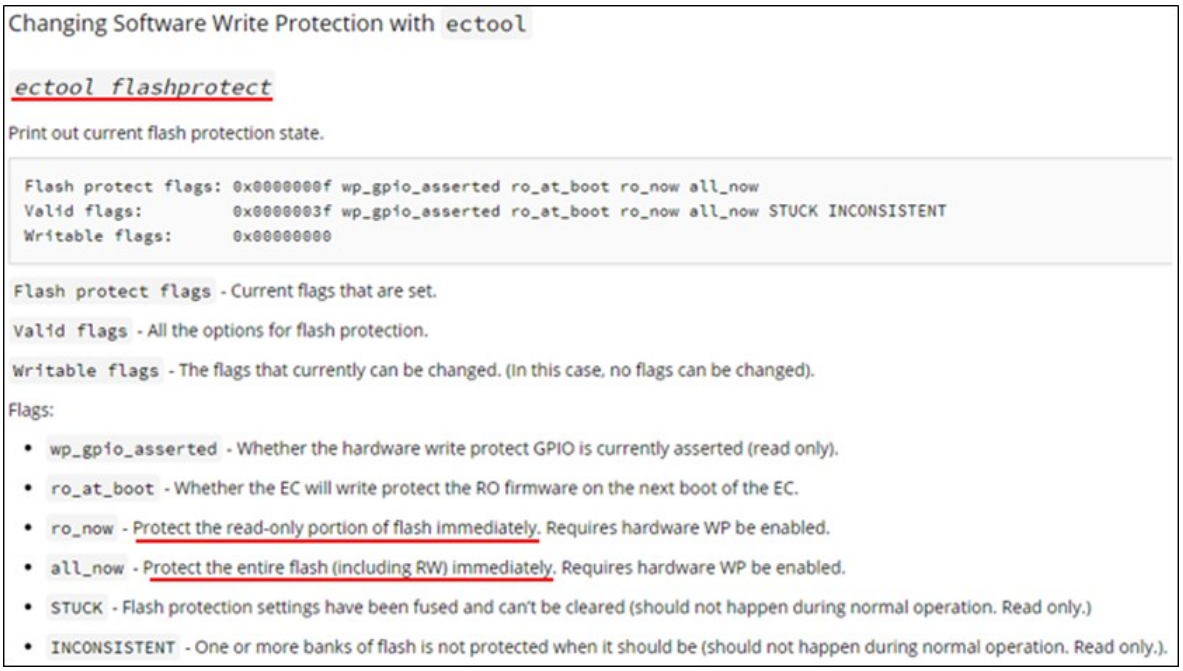


Source: https://chromium.googlesource.com/chromiumos/platform/ec/

83. The Lenovo 14e Chromebook and the Acer Chromebook C710 both include a first storage device for storing a first system software and a boot image. For example, the device includes flash memory ("first storage device") that stores the RO firmware ("boot image") and RW firmware ("first system software").



Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8__TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_111 (Slide 2).



Changing Software Write Protection with `ectool`

`ectool flashprotect`

Print out current flash protection state.

```
Flash protect flags: 0x0000000f wp_gpio_asserted ro_at_boot ro_now all_now
Valid flags:         0x0000003f wp_gpio_asserted ro_at_boot ro_now all_now STUCK INCONSISTENT
Writable flags:      0x00000000
```

`Flash protect flags` - Current flags that are set.

`Valid flags` - All the options for flash protection.

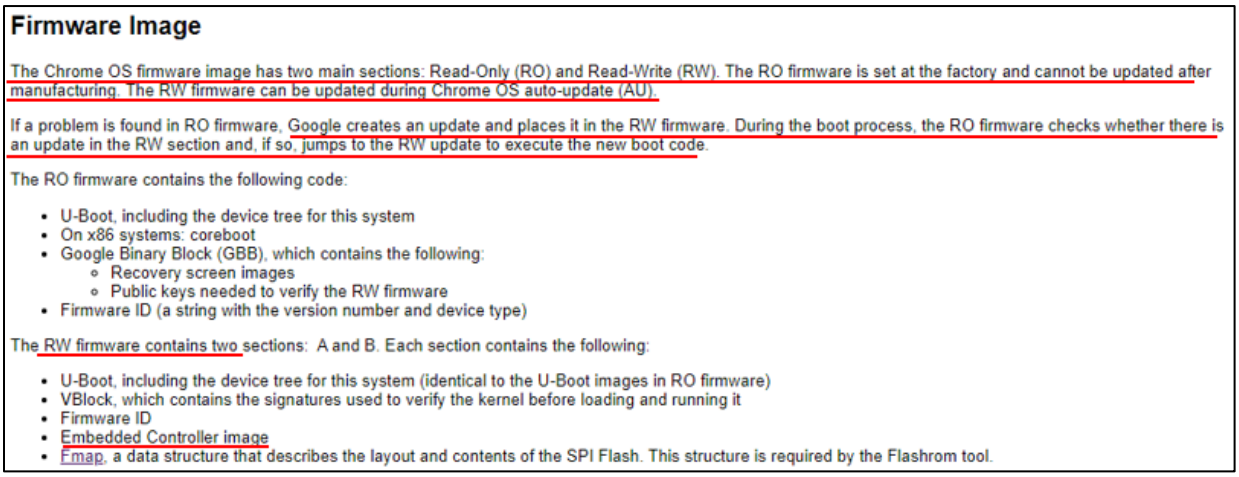
`Writable flags` - The flags that currently can be changed. (In this case, no flags can be changed).

Flags:

- `wp_gpio_asserted` - Whether the hardware write protect GPIO is currently asserted (read only).
- `ro_at_boot` - Whether the EC will write protect the RO firmware on the next boot of the EC.
- `ro_now` - Protect the read-only portion of flash immediately. Requires hardware WP be enabled.
- `all_now` - Protect the entire flash (including RW) immediately. Requires hardware WP be enabled.
- `STUCK` - Flash protection settings have been fused and can't be cleared (should not happen during normal operation. Read only.)
- `INCONSISTENT` - One or more banks of flash is not protected when it should be (should not happen during normal operation. Read only.).

Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+/master/docs/write_protection.md



Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
  - Recovery screen images
  - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

84. The RO firmware and RW firmware are stored in system memory.

BASE	SIZE	SECTION	DESCRIPTION
0x000000	0x200000	SI_ALL	Descriptor + ME
0x200000	0x0f0000	RW_SECTION_A	Read-Write Firmware A
0x2f0000	0x0f0000	RW_SECTION_B	Read-Write Firmware B
0x3e0000	0x010000	RW_MRC_CACHE	Memory Training Cache
0x3f0000	0x004000	RW_ELOG	Event Log
0x3f4000	0x004000	RW_SHARED	Shared Data
0x3f8000	0x002000	RW_VPD	Read-Write VPD
0x400000	0x200000	RW_LEGACY	Legacy Firmware
0x600000	0x004000	RO_VPD	Read-Only VPD
0x610000	0x000800	FMAP	Flash Map
0x610800	0x000040	RO_FRID	RO Firmware ID
0x611000	0x0ef000	GBB	Google Binary Block
0x700000	0x100000	BOOT_STUB	Read-Only Firmware

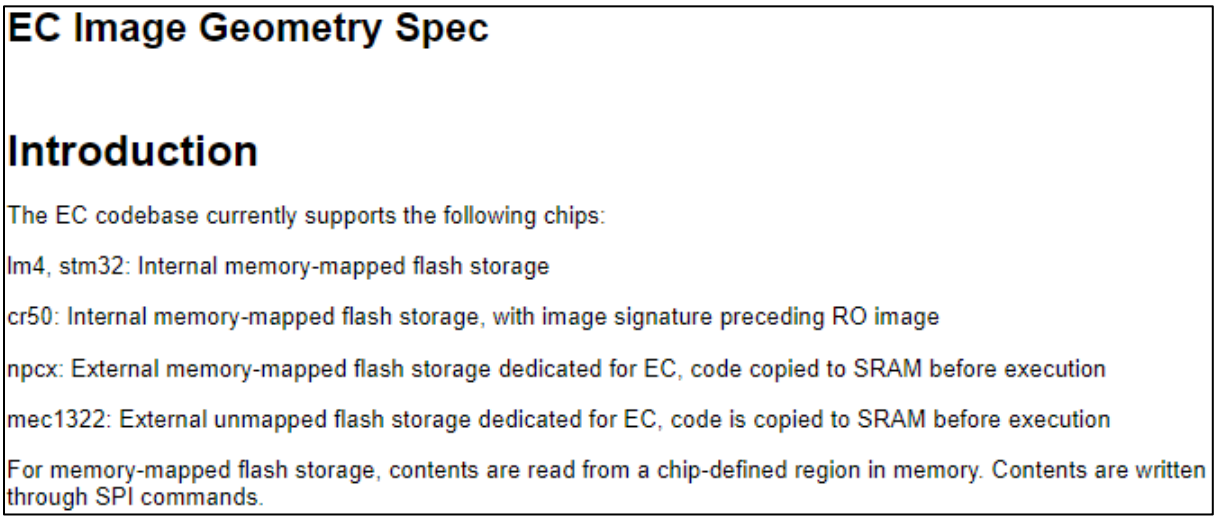
Source: https://docs.google.com/presentation/d/1h-nsDGlQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_1128 (Slide 33).

85. The Lenovo 14e Chromebook and the Acer Chromebook C710 both include a micro-controller that is coupled to the first storage device for respectively transforming the first system software and the boot image into system code and boot code. The micro-controller orderly executes the boot code and the system code to control booting of the embedded system.

86. For example, the EC ("micro-controller") is connected to flash memory ("first storage device") that stores both the RO firmware ("boot image") and RW firmware ("first system software"). The firmware can have different configurations, which relates to how it uses flash and RAM memory for system code and boot code execution. For example, firmware images in Chrome OS support configurations like Internal Mapped Storage, Code Copied to RAM for Use, etc.

87. During system boot, linker scripts provide information as to how different sections of the RO firmware ("boot image") and RW firmware ("first system software") map to

different sections of memory. The system provides linker scripts for both the RO firmware and the RW firmware in all the supported memory configurations. Before booting, a small program called Boot Loader (i.e., start-up code) uses linker scripts to load sections of the RO firmware and RW firmware into different parts of memory. The EC executes the mapped RO firmware ("boot code") and the mapped RW firmware ("system code") only after they are loaded into specified sections of memory determined by linker scripts. Accordingly, the RO firmware ("boot image") and RW firmware ("first system software") are transformed into memory-mapped executable code (i.e., "boot code" and "first system code") after being mapped and loaded into their respective sections of memory.



EC Image Geometry Spec

Introduction

The EC codebase currently supports the following chips:

lm4, stm32: Internal memory-mapped flash storage

cr50: Internal memory-mapped flash storage, with image signature preceding RO image

npcx: External memory-mapped flash storage dedicated for EC, code copied to SRAM before execution

mec1322: External unmapped flash storage dedicated for EC, code is copied to SRAM before execution

For memory-mapped flash storage, contents are read from a chip-defined region in memory. Contents are written through SPI commands.

Source: https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec

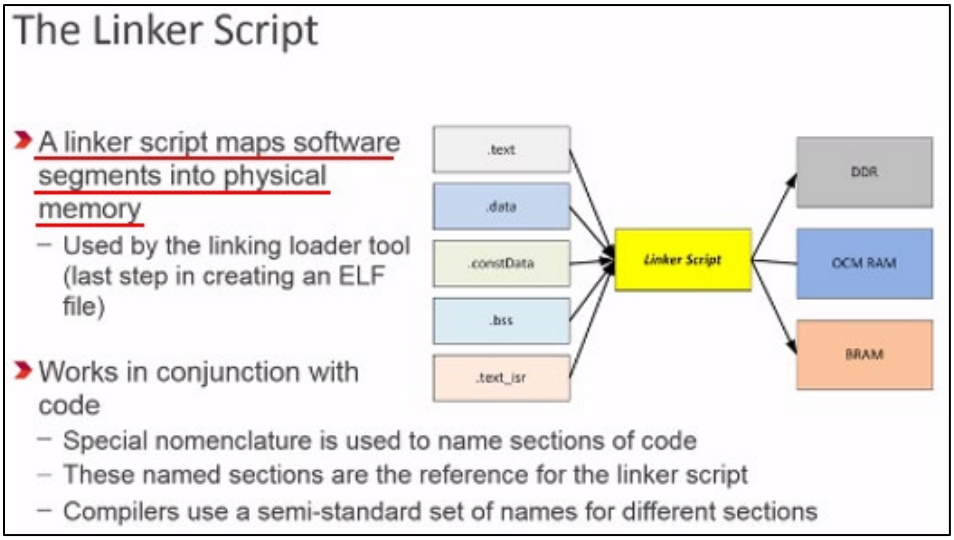
Supported Configurations

With the changes proposed here, we aim to support ECs with the following configurations:

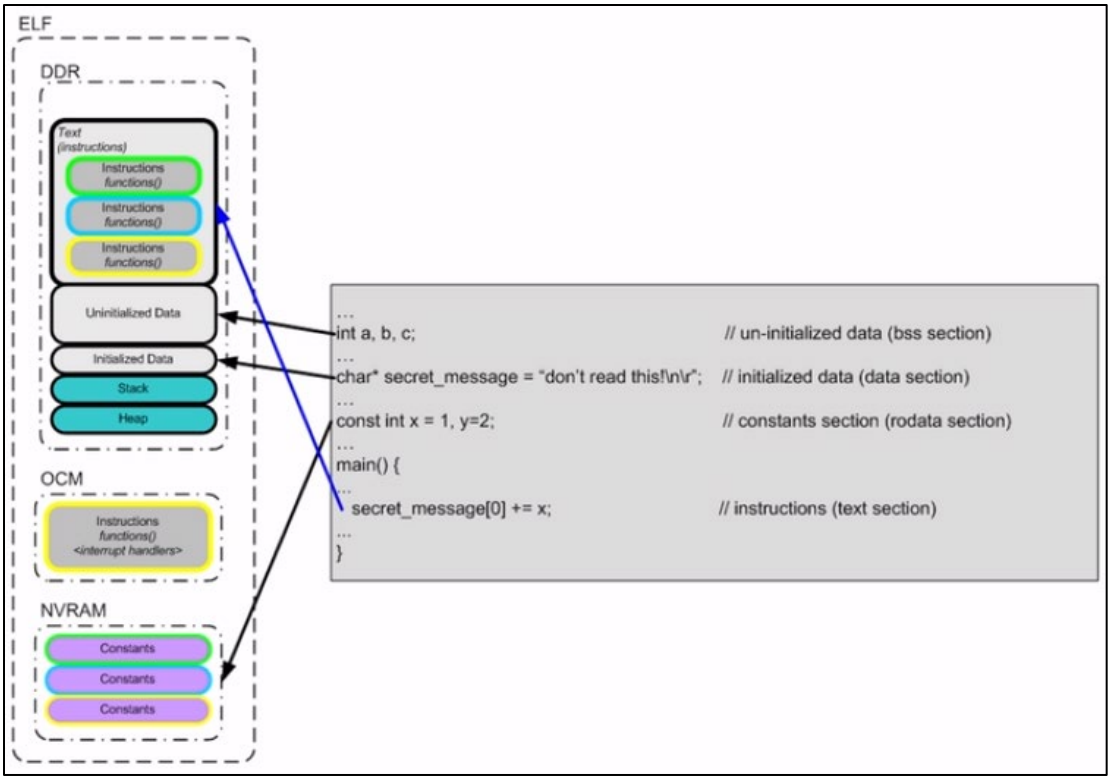
- Internal mapped storage
- External mapped storage (shared or dedicated)
- External unmapped storage (shared or dedicated)
- Code executed directly from mapped storage
- Code copied to SRAM before use

- One RO image
- One contiguous region of storage belonging to the EC, containing the RO image, that can be write protected
- Up to one RW image
- Up to one contiguous region of storage belonging to the EC, containing the RW image, that will not be write protected
- Support for any number of chip-specific non-image data pieces (*NIDs*) as part of the storage regions (loaders, headers, etc - any piece of data on EC storage that isn't part of the RO or RW image)
- Write protected and writable EC storage regions need not be mutually contiguous

Source: https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec



Source: https://www.xilinx.com/training/customer-training/using-linker-scripts.html (2:20)



Source: https://www.xilinx.com/training/customer-training/using-linker-scripts.html (9:52)

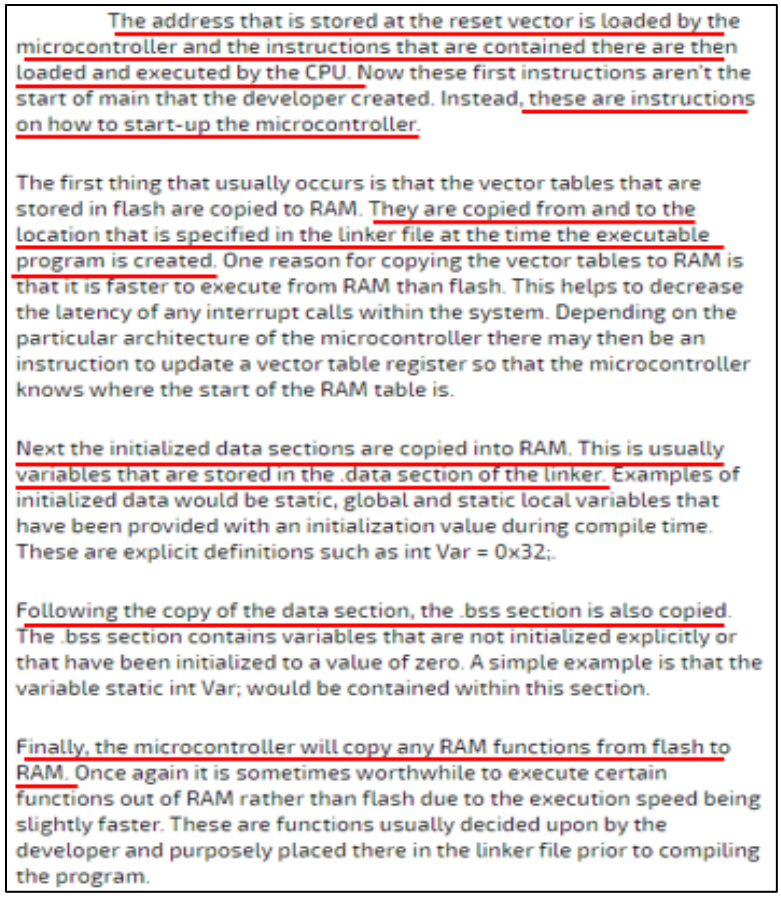
Linking

Our linker scripts were originally written to support internal memory-mapped storage with only RO and RW images (no NIDs). The scripts will be extended to support the new configs defined above. For example, the linker will produce a unified image with the RW image at the appropriate offset, as defined by CONFIG_RW_STORAGE_OFF / CONFIG_RW_STORAGE_SIZE.

Source: https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec

The microcontroller boot process starts by simply applying power to the system. Once the voltage rails stabilize, the microcontroller looks to the reset vector for the location in flash where the start-up instruction can be found. The reset vector is a special location within the flash memory

Source: https://www.beningo.com/understanding-the-microcontroller-boot-process/



The address that is stored at the reset vector is loaded by the microcontroller and the instructions that are contained there are then loaded and executed by the CPU. Now these first instructions aren't the start of main that the developer created. Instead, these are instructions on how to start-up the microcontroller.

The first thing that usually occurs is that the vector tables that are stored in flash are copied to RAM. They are copied from and to the location that is specified in the linker file at the time the executable program is created. One reason for copying the vector tables to RAM is that it is faster to execute from RAM than flash. This helps to decrease the latency of any interrupt calls within the system. Depending on the particular architecture of the microcontroller there may then be an instruction to update a vector table register so that the microcontroller knows where the start of the RAM table is.

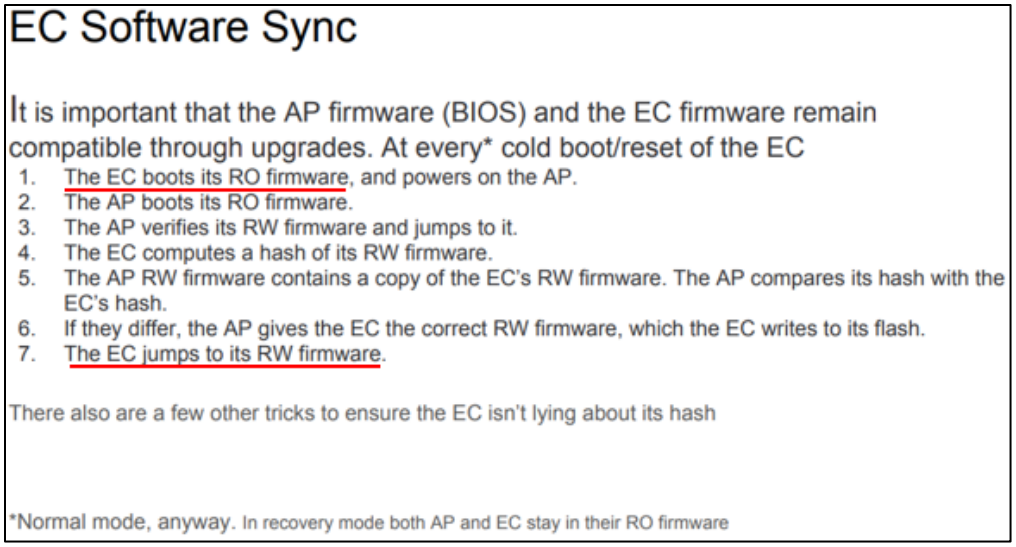
Next the initialized data sections are copied into RAM. This is usually variables that are stored in the .data section of the linker. Examples of initialized data would be static, global and static local variables that have been provided with an initialization value during compile time. These are explicit definitions such as int Var = 0x32;.

Following the copy of the data section, the .bss section is also copied. The .bss section contains variables that are not initialized explicitly or that have been initialized to a value of zero. A simple example is that the variable static int Var; would be contained within this section.

Finally, the microcontroller will copy any RAM functions from flash to RAM. Once again it is sometimes worthwhile to execute certain functions out of RAM rather than flash due to the execution speed being slightly faster. These are functions usually decided upon by the developer and purposely placed there in the linker file prior to compiling the program.

Source: https://www.beningo.com/understanding-the-microcontroller-boot-process/

88. During initialization and booting of the EC ("micro-controller"), the executable version of the RO firmware ("boot code") is executed first. The RO firmware then runs the executable version of the RW firmware ("first system code"). Accordingly, the boot code and system code are orderly executed to control booting of the embedded system.



EC Software Sync

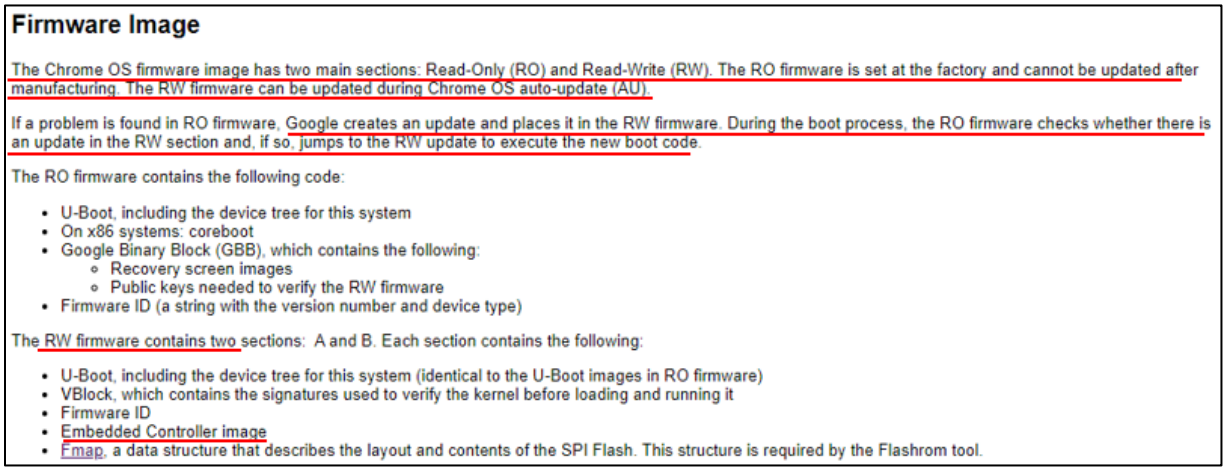
It is important that the AP firmware (BIOS) and the EC firmware remain compatible through upgrades. At every* cold boot/reset of the EC

1. The EC boots its RO firmware, and powers on the AP.
2. The AP boots its RO firmware.
3. The AP verifies its RW firmware and jumps to it.
4. The EC computes a hash of its RW firmware.
5. The AP RW firmware contains a copy of the EC's RW firmware. The AP compares its hash with the EC's hash.
6. If they differ, the AP gives the EC the correct RW firmware, which the EC writes to its flash.
7. The EC jumps to its RW firmware.

There also are a few other tricks to ensure the EC isn't lying about its hash

*Normal mode, anyway. In recovery mode both AP and EC stay in their RO firmware

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).



Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
  - Recovery screen images
  - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

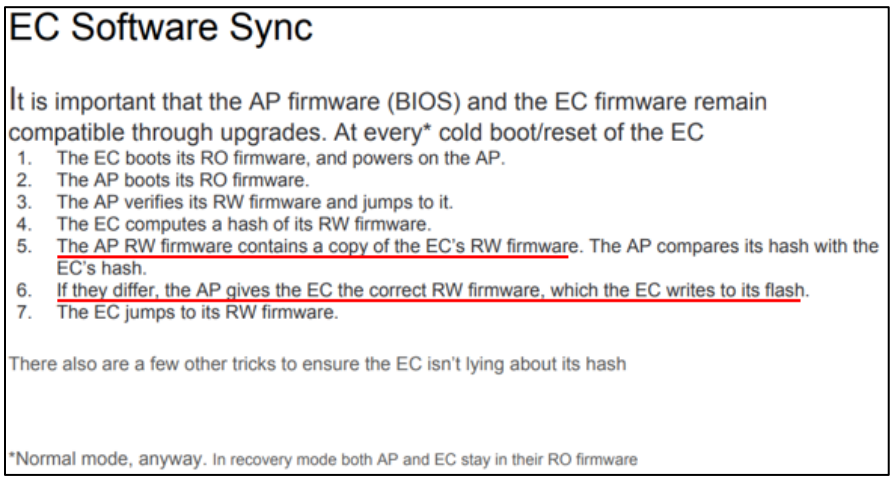
BASE	SIZE	SECTION	DESCRIPTION
0x000000	0x200000	SI_ALL	Descriptor + ME
0x200000	0x0f0000	RW_SECTION_A	Read-Write Firmware A
0x2f0000	0x0f0000	RW_SECTION_B	Read-Write Firmware B
0x3e0000	0x010000	RW_MRC_CACHE	Memory Training Cache
0x3f0000	0x004000	RW_ELOG	Event Log
0x3f4000	0x004000	RW_SHARED	Shared Data
0x3f8000	0x002000	RW_VPD	Read-Write VPD
0x400000	0x200000	RW_LEGACY	Legacy Firmware
0x600000	0x004000	RO_VPD	Read-Only VPD
0x610000	0x000800	FMAP	Flash Map
0x610800	0x000040	RO_FRID	RO Firmware ID
0x611000	0x0ef000	GBB	Google Binary Block
0x700000	0x100000	BOOT_STUB	Read-Only Firmware

Source: https://docs.google.com/presentation/d/1h-nsDGlQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_1128 (Slide 33).

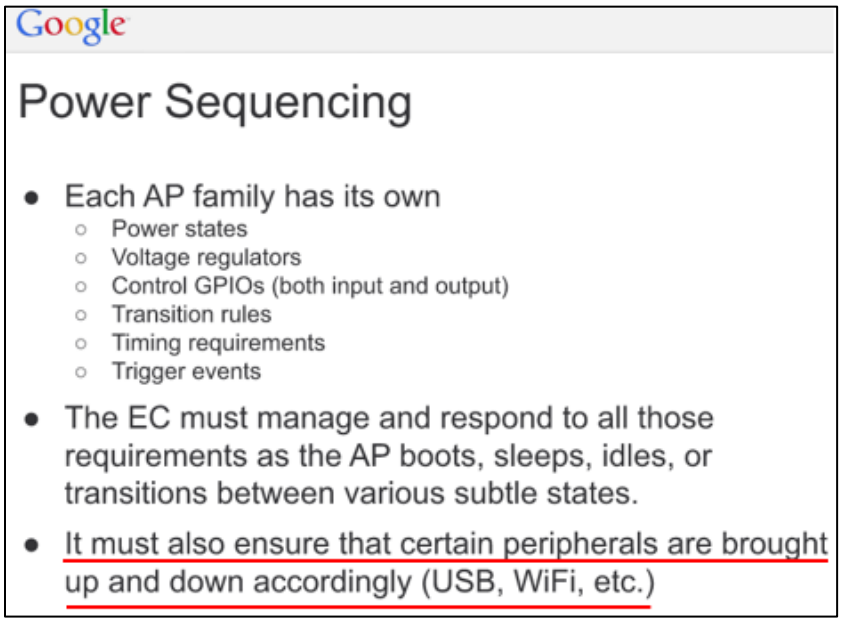
89. The Lenovo 14e Chromebook and the Acer Chromebook C710 both include a connecting interface that is coupled to the micro-controller and further coupled to an external data storage device through a data transmission media. The external data storage device stores a second system software.

90. For example, the EC ("micro-controller"), which controls peripheral connections of the device like USB, Wi-Fi, etc., receives an updated version of the RW firmware ("second system software") from the device's Application Processor (AP). Specifically, the updated RW firmware ("second system software") is read from the network server when the device is connected to the internet through a Wi-Fi network interface. Accordingly, the EC ("micro-controller") is configured to be coupled to the network server ("external data storage device") through the Wi-Fi network interface ("connecting interface"). The network server stores the RW

firmware update ("second system software"), which can be read by the device via the internet ("data transmission media").



Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).



Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8__TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_142 (Slide 21).

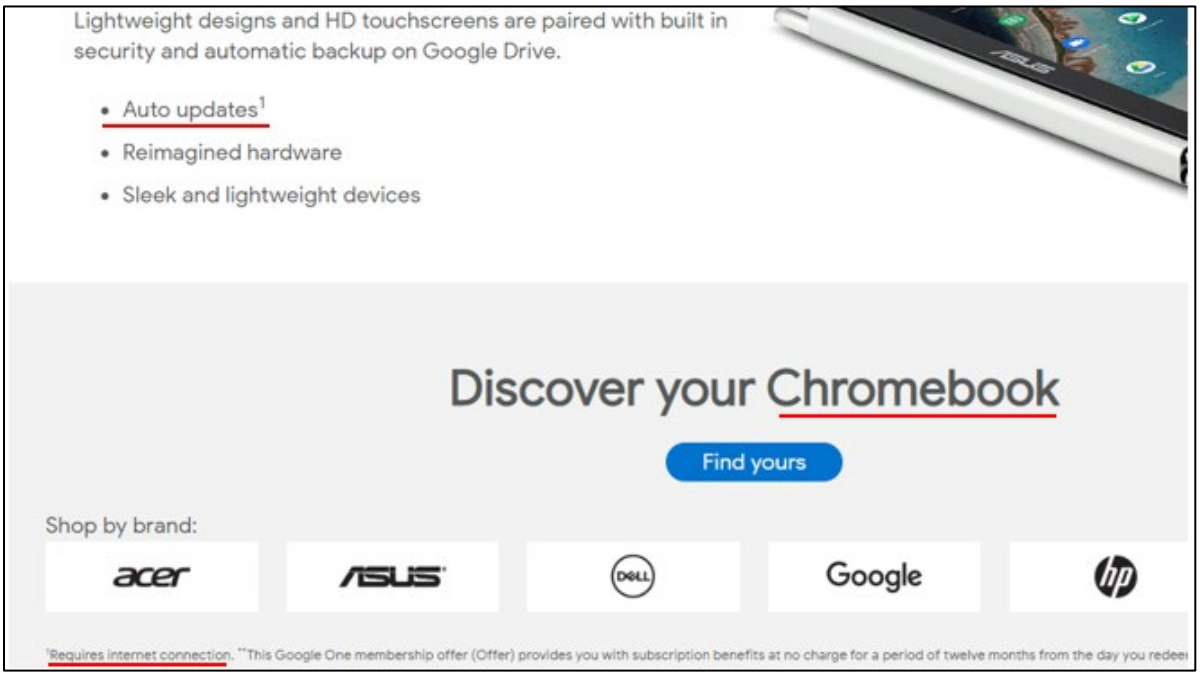
3. Verified Boot

Chromebook's startup is very different from Windows or Mac machines. When Chrome OS boots, it compares every component of the operating system with the current version verified by Google through the Internet. If there is a discrepancy, it will replace with the up-to-date version. Every time the Chromebook starts up, it does the self-check called "Verified Boot."

The self-check ensures Chrome OS in the right shape; it plays a fundamental role in Chromebook security mechanism.

- Drive automatic update: download new updates of Chrome OS when Verified Boot;
- Repair corrupted OS: take Chrome OS back if malware manages to escape the Sandbox;

Source: https://www.keepds.com/tool/list?os=c



Source: https://www.walmart.ca/en/electronics/laptops-computers/laptops-notebooks/chromebooks/google/N-1990+1000268?mtr=mdv_00439&icid=electronics_wmg_display_walmart_l4hb_wk16_google_chromebook_en

> Google rolls out security updates as soon as they're ready and applies them when a Chromebook boots up. In other words, during the boot sequence, Chrome OS checks to see if a new update is available. If yes, installs it without interrupting the user. At this point, your Internet has to be available.

Source: https://www.keepds.com/tool/list?os=c

> If peer-to-peer (P2P) networking is available, devices can automatically update Chrome from nearby devices of the same model. This option reduces external network traffic. If P2P automatic updating fails or isn't possible on your network, devices update as usual. They either download the update from Google's servers or an intermediate web-caching proxy server.

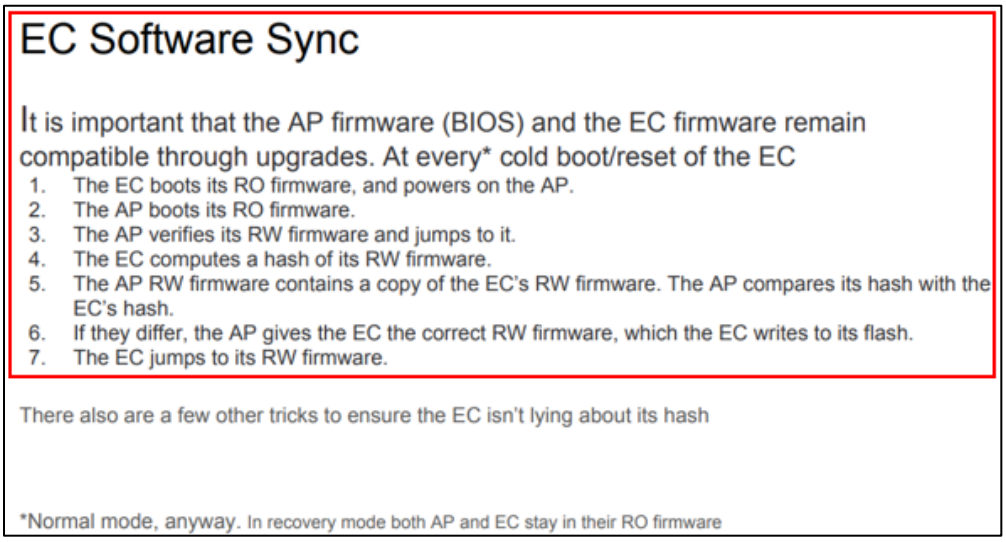
Source: https://support.google.com/chrome/a/answer/3168106?hl=en

> **Firmware Image**
>
> The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).
>
> If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.
>
> The RO firmware contains the following code:
>
> - U-Boot, including the device tree for this system
> - On x86 systems: coreboot
> - Google Binary Block (GBB), which contains the following:
> - Recovery screen images
> - Public keys needed to verify the RW firmware
> - Firmware ID (a string with the version number and device type)
>
> The RW firmware contains two sections: A and B. Each section contains the following:
>
> - U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
> - VBlock, which contains the signatures used to verify the kernel before loading and running it
> - Firmware ID
> - Embedded Controller image
> - Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

91. The Lenovo 14e Chromebook and the Acer Chromebook C710 each includes boot code, which includes update agent interface programming (UAIP). The micro-controller is capable of executing the update agent interface programming to read the second system software from the external data storage device through the connecting interface before executing the system code.

92. During the boot process of the system, the executable version of the RO firmware ("boot code") is executed before the RW firmware. And before the RW firmware ("first system code") is executed, the EC initiates a Software Sync process by powering on and booting the device's AP. The system verifies and if needed, updates the RW firmware by comparing the version of the RW firmware. If there is an available update to the RW firmware ("second system software"), the update is sent to flash memory. Accordingly, the RO firmware ("boot code") includes code or programming ("update agent interface programming") to initiate the software sync process that updates the RW firmware.



EC Software Sync

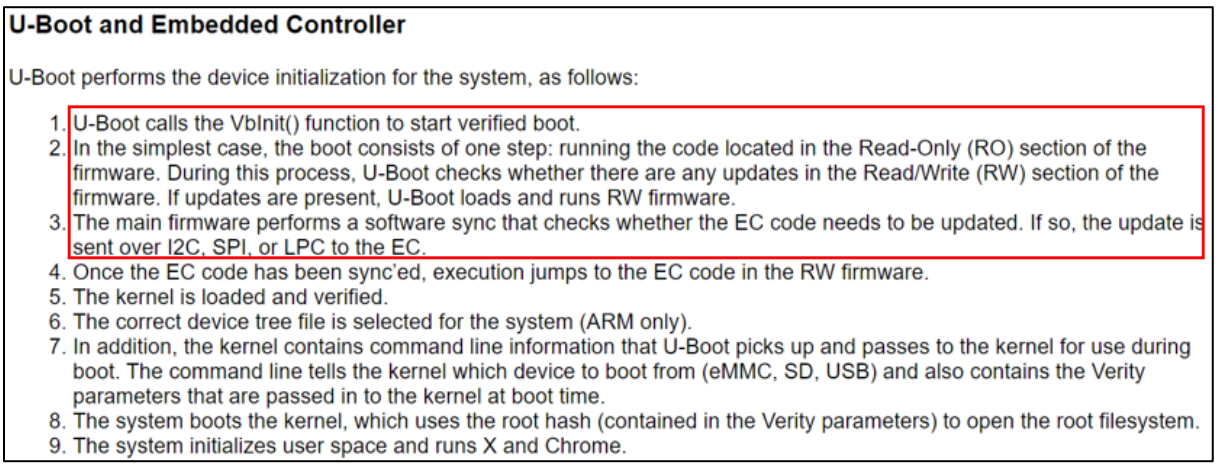
It is important that the AP firmware (BIOS) and the EC firmware remain compatible through upgrades. At every* cold boot/reset of the EC

1. The EC boots its RO firmware, and powers on the AP.
2. The AP boots its RO firmware.
3. The AP verifies its RW firmware and jumps to it.
4. The EC computes a hash of its RW firmware.
5. The AP RW firmware contains a copy of the EC's RW firmware. The AP compares its hash with the EC's hash.
6. If they differ, the AP gives the EC the correct RW firmware, which the EC writes to its flash.
7. The EC jumps to its RW firmware.

There also are a few other tricks to ensure the EC isn't lying about its hash

*Normal mode, anyway. In recovery mode both AP and EC stay in their RO firmware

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).

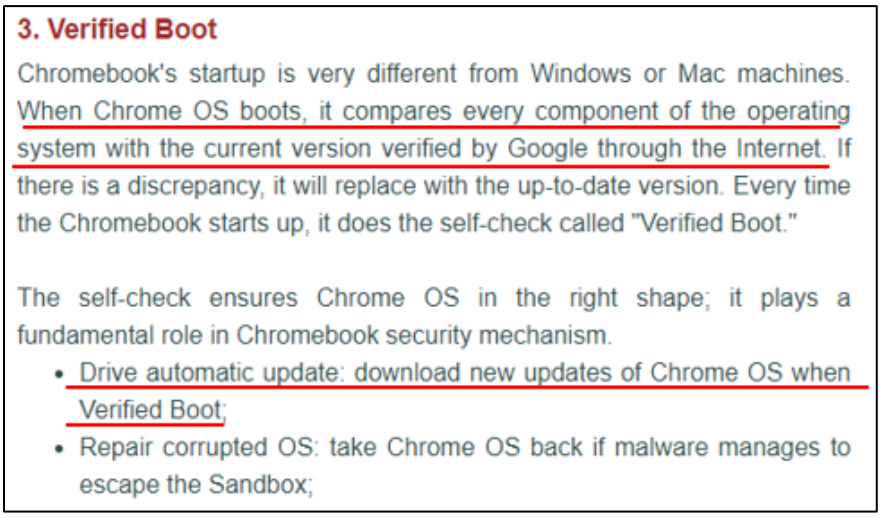


U-Boot and Embedded Controller

U-Boot performs the device initialization for the system, as follows:

1. U-Boot calls the VbInit() function to start verified boot.
2. In the simplest case, the boot consists of one step: running the code located in the Read-Only (RO) section of the firmware. During this process, U-Boot checks whether there are any updates in the Read/Write (RW) section of the firmware. If updates are present, U-Boot loads and runs RW firmware.
3. The main firmware performs a software sync that checks whether the EC code needs to be updated. If so, the update is sent over I2C, SPI, or LPC to the EC.
4. Once the EC code has been sync'ed, execution jumps to the EC code in the RW firmware.
5. The kernel is loaded and verified.
6. The correct device tree file is selected for the system (ARM only).
7. In addition, the kernel contains command line information that U-Boot picks up and passes to the kernel for use during boot. The command line tells the kernel which device to boot from (eMMC, SD, USB) and also contains the Verity parameters that are passed in to the kernel at boot time.
8. The system boots the kernel, which uses the root hash (contained in the Verity parameters) to open the root filesystem.
9. The system initializes user space and runs X and Chrome.

Source: https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1



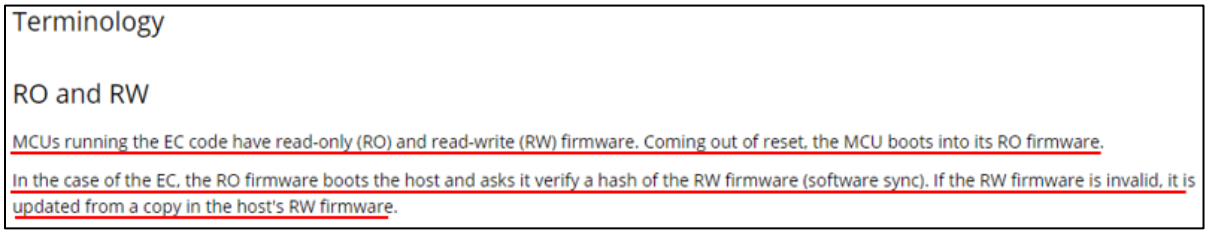
3. Verified Boot

Chromebook's startup is very different from Windows or Mac machines. When Chrome OS boots, it compares every component of the operating system with the current version verified by Google through the Internet. If there is a discrepancy, it will replace with the up-to-date version. Every time the Chromebook starts up, it does the self-check called "Verified Boot."

The self-check ensures Chrome OS in the right shape; it plays a fundamental role in Chromebook security mechanism.

- Drive automatic update: download new updates of Chrome OS when Verified Boot;
- Repair corrupted OS: take Chrome OS back if malware manages to escape the Sandbox;

Source: https://www.keepds.com/tool/list?os=c



Terminology

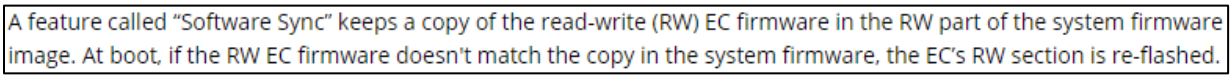
RO and RW

MCUs running the EC code have read-only (RO) and read-write (RW) firmware. Coming out of reset, the MCU boots into its RO firmware.

In the case of the EC, the RO firmware boots the host and asks it verify a hash of the RW firmware (software sync). If the RW firmware is invalid, it is updated from a copy in the host's RW firmware.

Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+/master/docs/write_protection.md

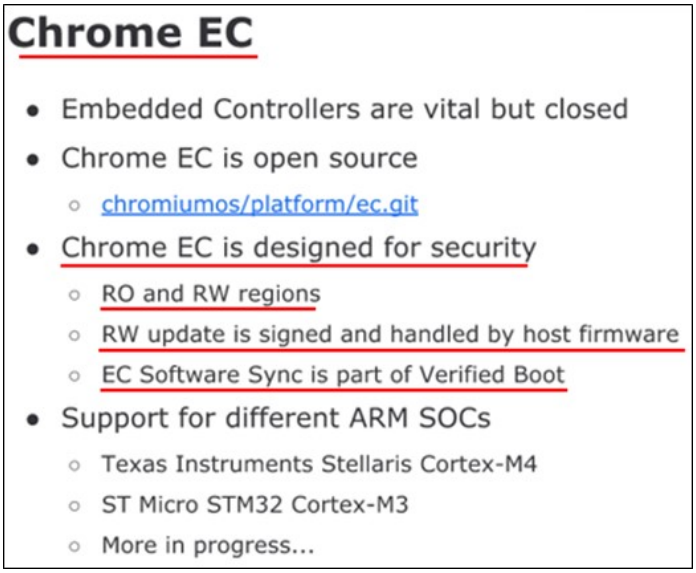


A feature called "Software Sync" keeps a copy of the read-write (RW) EC firmware in the RW part of the system firmware image. At boot, if the RW EC firmware doesn't match the copy in the system firmware, the EC's RW section is re-flashed.

Source: https://chromium.googlesource.com/chromiumos/platform/ec/

93. During the Verified Boot process, the system checks for updates to the RW firmware. The RO firmware includes programming that verifies and compares the version of the RW firmware on the device with the version of the RW firmware on the network server. The RO firmware further includes programming that updates the RW firmware based on that

comparison. The Software Sync process is a part of the Verified Boot process. It includes programming that verifies and compares version of the RW firmware.



Source: https://docs.google.com/presentation/d/1h-nsDGlQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298 (Slide 29).

```
VbError_t VbEcSoftwareSync(VbCommonParams *cparams)
{
        VbSharedDataHeader *shared =
                (VbSharedDataHeader *)cparams->shared_data_blob;
        int in_rw = 0;
        int rv;
        const uint8_t *ec_hash = NULL;
        int ec_hash_size;
        const uint8_t *rw_hash = NULL;
        int rw_hash_size;
        const uint8_t *expected = NULL;
        int expected_size;
        uint8_t expected_hash[SHA256_DIGEST_SIZE];
        int need_update = 0;
        int i;

        /* Determine whether the EC is in RO or RW */
        rv = VbExEcRunningRW(&in_rw);
```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```
/* Get hash of EC-RW */
rv = VbExEcHashRW(&ec_hash, &ec_hash_size);
if (rv) {
        VBDEBUG(("VbEcSoftwareSync() - "
                 "VbExEcHashRW() returned %d\n", rv));
        VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_FAILED);
        return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
}
if (ec_hash_size != SHA256_DIGEST_SIZE) {
        VBDEBUG(("VbEcSoftwareSync() - "
                 "VbExEcHashRW() says size %d, not %d\n",
                 ec_hash_size, SHA256_DIGEST_SIZE));
        VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_SIZE);
        return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
}

VBDEBUG(("EC hash:"));
for (i = 0; i < SHA256_DIGEST_SIZE; i++)
        VBDEBUG(("%02x", ec_hash[i]));
VBDEBUG(("\n"));
```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```
/*
 * Get expected EC-RW image if we're sure we need to update (because the
 * expected hash didn't match the EC) or we still don't know (because
 * there was no expected hash and we need the image to compute one
 * ourselves).
 */
if (need_update || !rw_hash) {
        /* Get expected EC-RW image */
        rv = VbExEcGetExpectedRW(shared->firmware_index ?
                                 VB_SELECT_FIRMWARE_B :
                                 VB_SELECT_FIRMWARE_A,
                                 &expected, &expected_size);
        if (rv) {
                VBDEBUG(("VbEcSoftwareSync() - "
                         "VbExEcGetExpectedRW() returned %d\n", rv));
                VbSetRecoveryRequest(VBNV_RECOVERY_EC_EXPECTED_IMAGE);
                return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
        }
        VBDEBUG(("VbEcSoftwareSync() - expected len = %d\n",
                 expected_size));

        /* Hash expected image */
        internal_SHA256(expected, expected_size, expected_hash);
        VBDEBUG(("Computed hash of expected image:"));
        for (i = 0; i < SHA256_DIGEST_SIZE; i++)
                VBDEBUG(("%02x", expected_hash[i]));
        VBDEBUG(("\n"));
```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```
/*
 * We need to update, but the expected EC image doesn't match
 * the expected EC hash we were given.
 */
VBDEBUG(("VbEcSoftwareSync() - "
         "VbExEcGetExpectedRW() returned %d\n", rv));
VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_MISMATCH);
return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
```

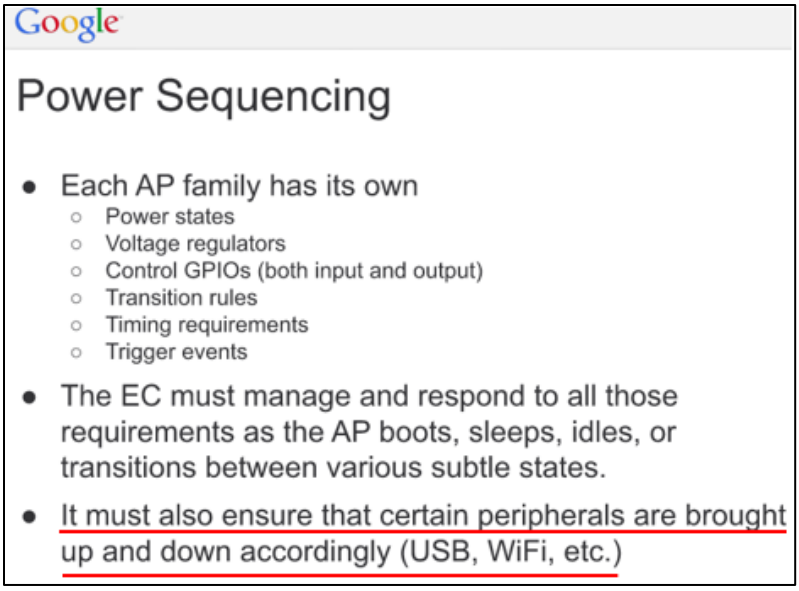
Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```
/* Update EC if necessary */
if (need_update) {
        VBDEBUG(("VbEcSoftwareSync() updating EC-RW...\n"));

        if (shared->flags & VBSD_EC_SLOW_UPDATE) {
                VBDEBUG(("VbEcSoftwareSync() - "
                         "EC is slow. Show WAIT screen.\n"));
                /*
                 * FIXME(crosbug.com/p/12257): Ensure the VGA Option
                 * ROM is loaded!
                 */
                VbDisplayScreen(cparams, VB_SCREEN_WAIT, 0, &vnc);
        }

        rv = VbExEcUpdateRW(expected, expected_size);
        if (rv == VBERROR_EC_REBOOT_TO_RO_REQUIRED) {
                /*
                 * Reboot required.  May need to unprotect RW before
                 * updating, or may need to reboot after RW updated.
                 * Either way, it's not an error requiring recovery
                 * mode.
                 */
                VBDEBUG(("VbEcSoftwareSync() - "
                         "VbExEcUpdateRW() needs reboot\n"));
                return rv;
```

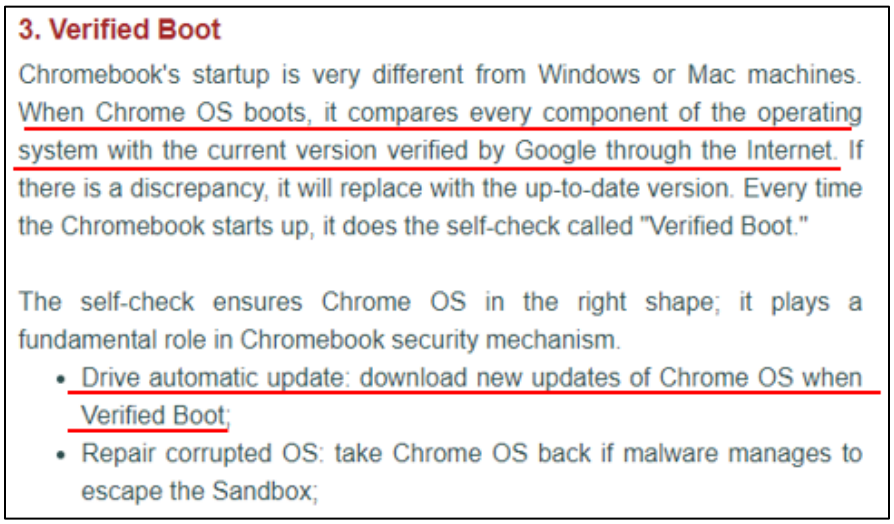
Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

94. The EC controls peripheral connections like USB, Wi-Fi, etc. During the Verified Boot process, the device receives updates of the RW firmware from the network server ("external data storage device") when connected to the internet through Wi-Fi, for example. The RW firmware is updated during the Verified Boot process.

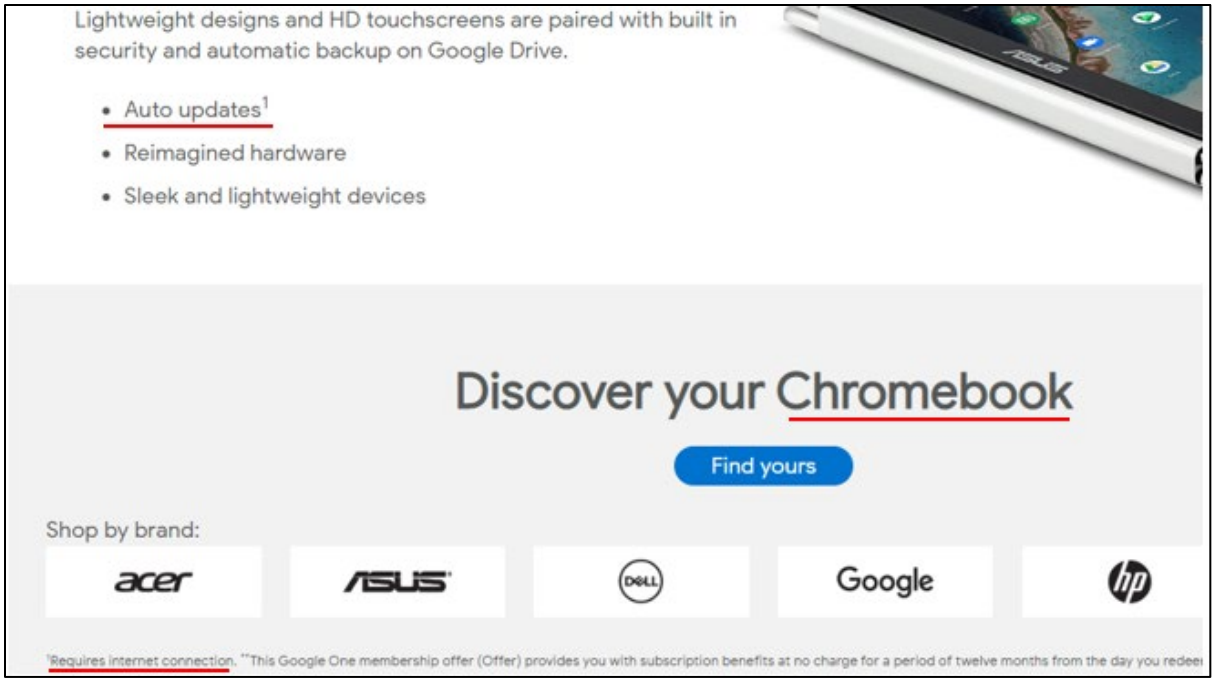
95. The EC initiates the process of updating the RW firmware by activating the Wi-Fi network interface, connecting to the internet, and starting the Software Sync process. Accordingly, the system includes code or programming ("update agent interface programming") to read the RW firmware update ("second system software") from the network server ("external data storage device") before executing the current version of the RW firmware ("first system code").



Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8__TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_142 (Slide 21).



Source: https://www.keepds.com/tool/list?os=c



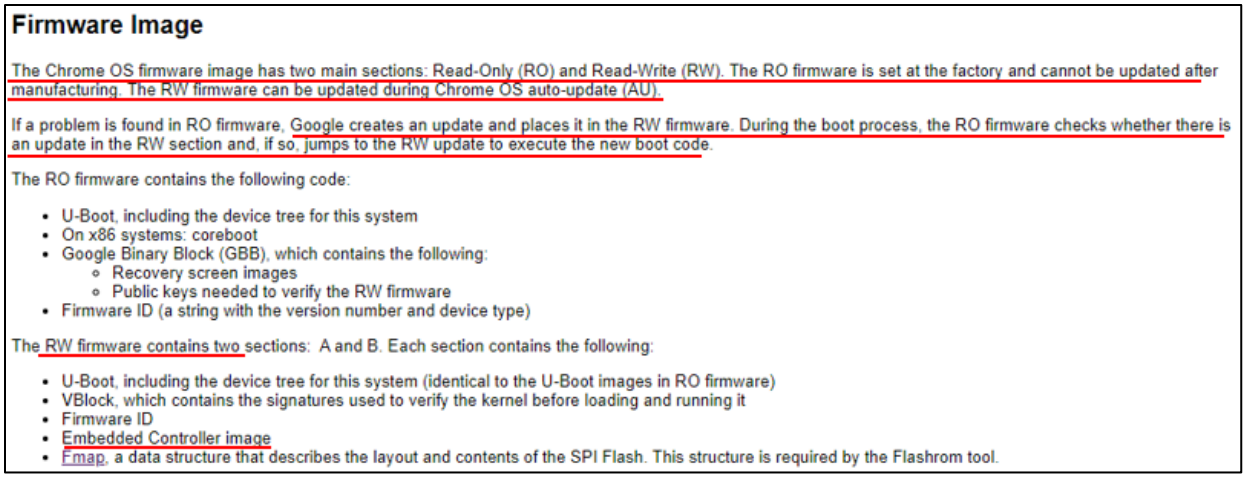
Source: https://www.walmart.ca/en/electronics/laptops-computers/laptops-notebooks/chromebooks/google/N-1990+1000268?mtr=mdv_00439&icid=electronics_wmg_display_walmart_l4hb_wk16_google_chromebook_en

> Google rolls out security updates as soon as they're ready and applies them when a Chromebook boots up. In other words, during the boot sequence, Chrome OS checks to see if a new update is available. If yes, installs it without interrupting the user. At this point, your Internet has to be available.

Source: https://www.keepds.com/tool/list?os=c

> If peer-to-peer (P2P) networking is available, devices can automatically update Chrome from nearby devices of the same model. This option reduces external network traffic. If P2P automatic updating fails or isn't possible on your network, devices update as usual. They either download the update from Google's servers or an intermediate web-caching proxy server.

Source: https://support.google.com/chrome/a/answer/3168106?hl=en



Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
  - Recovery screen images
  - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

96. Defendants Lenovo and Compal have had knowledge of the ’612 Patent at least as of the date when they were notified of the filing of this action.

97. Liberty Patents has been damaged as a result of the infringing conduct by Defendants alleged above. Thus, Defendants are liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

98. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the ’612 Patent.

ADDITIONAL ALLEGATIONS REGARDING INFRINGEMENT

99. Defendants Lenovo and Compal have also indirectly infringed the ’959 Patent, the ’573 Patent, and the ’612 Patent by inducing others to directly infringe the ’959 Patent, the ’573 Patent, and the ’612 Patent. Defendants Lenovo and Compal have induced the end-users, their customers, to directly infringe (literally and/or under the doctrine of equivalents) the ’959 Patent, the ’573 Patent, and the ’612 Patent by using the accused products.

100. Defendants Lenovo and Compal took active steps, directly and/or through contractual relationships with others, with the specific intent to cause them to use the accused products in a manner that infringes one or more claims of the patents-in-suit, including, for example, claim 1 of the '959 Patent, claim 13 of the '573 Patent, and claim 1 of the '612 Patent.

101. Such steps by Defendants Lenovo and Compal included, among other things, advising or directing customers and end-users to use the accused products in an infringing manner; advertising and promoting the use of the accused products in an infringing manner; and/or distributing instructions that guide users to use the accused products in an infringing manner.

102. Defendants Lenovo and Compal performed these steps, which constitute induced infringement, with the knowledge of the '959 Patent, the '573 Patent, and the '612 Patent and with the knowledge that the induced acts constitute infringement.

103. Defendants Lenovo and Compal were and are aware that the normal and customary use of the accused products by their customers would infringe the '959 Patent, the '573 Patent, and the '612 Patent. The inducement by Defendants Lenovo and Compal is ongoing.

104. Defendants Lenovo and Compal have also induced their affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on their or their affiliates' behalf, to directly infringe (literally and/or under the doctrine of equivalents) the '959 Patent, the '573 Patent, and the '612 Patent by importing, selling or offering to sell the accused products, including, for example, Allegiance Computer Technologies,[16] Amazon, Best Buy, CDW, Fry's Electronics, Micro Center, Office Depot, Staples, Target, Walmart, and others.

[16] *See, e.g.*, www.allegiancepc.com/index.php/computers-servers.

105. Defendants Lenovo and Compal have a significant role in placing the accused products in the stream of commerce with the expectation and knowledge that they will be purchased by consumers in Texas and elsewhere in the United States.

106. Defendants Lenovo and Compal purposefully direct or control the making of accused products and their shipment to the United States, using established distribution channels, for sale in Texas and elsewhere within the United States.

107. Defendants Lenovo and Compal purposefully direct or control the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. Defendants' established United States distribution channels include one or more United States based affiliates (e.g., Lenovo (United States) Inc. for Defendant Lenovo; Auscom Engineering Inc. and Bizcom Electronics, Inc. for Compal).

108. Defendants Lenovo and Compal purposefully direct or control the sale of the accused products online and in nationwide retailers, including for sale in Texas and elsewhere in the United States, and expect and intend that the accused products will be so sold.

109. Defendants Lenovo and Compal purposefully place the accused products—whether by themselves or through subsidiaries, affiliates, or third parties—into an international supply chain, knowing that the accused products will be sold in the United States, including Texas. Therefore, Defendants also facilitate the sale of the accused products in Texas.

110. Defendants Lenovo and Compal took active steps, directly and/or through contractual relationships with others,[17] with the specific intent to cause such persons to import, sell, or offer to sell the accused products in a manner that infringes one or more claims of the

[17] Compal has entered into Trading and Manufacturing Agreements with companies, such as Acer and Dell, that import, market, and sell laptop and notebook computers within the United States, including Texas. *See supra* Compal's 2019 Annual Report at 156.

patents-in-suit, including, for example, claim 1 of the '959 Patent, claim 13 of the '573 Patent, and claim 1 of the '612 Patent.

111. Such steps by Defendants Lenovo and Compal included, among other things, making or selling the accused products outside of the United States for importation into or sale in the United States, or knowing that such importation or sale would occur; and directing, facilitating, or influencing their affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on their behalf, to import, sell, or offer to sell the accused products in an infringing manner.

112. Defendants Lenovo and Compal performed these steps, which constitute induced infringement, with the knowledge of the '959 Patent, the '573 Patent, and the '612 Patent and with the knowledge that the induced acts would constitute infringement.

113. Defendants Lenovo and Compal performed such steps in order to profit from the eventual sale of the accused products in the United States.

114. The inducement by Defendants Lenovo and Compal is ongoing.

115. Defendants Lenovo and Compal have also indirectly infringed by contributing to the infringement of the '959 Patent, the '573 Patent, and the '612 Patent. Defendants Lenovo and Compal have contributed to the direct infringement of the '959 Patent, the '573 Patent, and the '612 Patent by the end-user of the accused products.

116. The accused products have special features that are specially designed to be used in an infringing way and that have no substantial uses other than ones that infringe the '959 Patent, the '573 Patent, and the '612 Patent, for example, claim 1 of the '959 Patent, claim 13 of the '573 Patent, and claim 1 of the '612 Patent.

117. The special features include, for example, executing computer instructions in an instruction cache used in a manner that infringes the ’959 Patent; power distribution and power management techniques used in a manner that infringes the ’573 Patent; and retrieving automatic software updates in an embedded system used in a manner that infringes the ’612 Patent.

118. These special features constitute a material part of the invention of one or more of the claims of the ’959 Patent, the ’573 Patent, and the ’612 Patent and are not staple articles of commerce suitable for substantial non-infringing use.

119. The contributory infringement by Defendants Lenovo and Compal is ongoing.

120. Defendants Lenovo and Compal had actual knowledge of the ’959 Patent, the ’573 Patent, and the ’612 Patent at least as of the date when they were notified of the filing of this action. Since at least that time, Defendants Lenovo and Compal have known the scope of the claims of the ’959 Patent, the ’573 Patent, and the ’612 Patent; the products that practice the ’959 Patent, the ’573 Patent, and the ’612 Patent; and that Liberty Patents is the owner of the ’959 Patent, the ’573 Patent, and the ’612 Patent.

121. By the time of trial, Defendants Lenovo and Compal will have known and intended (since receiving such notice) that their continued actions would infringe and actively induce and contribute to the infringement of one or more claims of the ’959 Patent, the ’573 Patent, and the ’612 Patent.

122. Furthermore, Defendants have a policy or practice of not reviewing the patents of others (including instructing their employees to not review the patents of others), and thus have been willfully blind of Liberty Patents’ patent rights. *See, e.g.*, M. Lemley, “Ignoring Patents,” 2008 Mich. St. L. Rev. 19 (2008).

123. Defendants' actions are at least objectively reckless as to the risk of infringing valid patents, and this objective risk was either known or should have been known by Defendants. Defendants Lenovo and Compal have knowledge of the '959 Patent, the '573 Patent, and the '612 Patent.

124. Defendants' customers have infringed the '959 Patent, the '573 Patent, and the '612 Patent, and Defendants have encouraged their customers' infringement.

125. The direct and indirect infringement of the '959 Patent, the '573 Patent, and the '612 Patent by Defendants Lenovo and Compal has been, and/or continues to be willful, intentional, deliberate, and/or in conscious disregard of Liberty Patents' rights under the patents-in-suit.

126. Liberty Patents has been damaged as a result of the infringing conduct by Defendants Lenovo and Compal alleged above. Thus, Defendants are liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

JURY DEMAND

Liberty Patents hereby requests a trial by jury on all issues so triable by right.

PRAYER FOR RELIEF

Liberty Patents requests that the Court find in its favor and against Lenovo and Compal, and that the Court grant Liberty Patents the following relief:

a. Judgment that one or more claims of the '959 Patent, the '573 Patent, and the '612 Patent have been infringed, either literally and/or under the doctrine of equivalents, by Defendants Lenovo and Compal and/or all others acting in concert therewith;

b. A permanent injunction enjoining Defendants Lenovo and Compal and their officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the '959 Patent, the '573 Patent, and the '612 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the '959 Patent, the '573 Patent, and the '612 Patent by such entities;

c. Judgment that Defendants Lenovo and Compal account for and pay to Liberty Patents all damages to and costs incurred by Liberty Patents because of Defendants' infringing activities and other conduct complained of herein, including an award of all increased damages to which Liberty Patents is entitled under 35 U.S.C. § 284;

d. That Liberty Patents be granted pre-judgment and post-judgment interest on the damages caused by Defendants' infringing activities and other conduct complained of herein;

e. That this Court declare this an exceptional case and award Liberty Patents its reasonable attorney's fees and costs in accordance with 35 U.S.C. § 285; and

f. That Liberty Patents be granted such other and further relief as the Court may deem just and proper under the circumstances.

Dated: September 28, 2020

Respectfully submitted,

/s/ *Zachariah S. Harrington*

Matthew J. Antonelli
Texas Bar No. 24068432
matt@ahtlawfirm.com
Zachariah S. Harrington
Texas Bar No. 24057886
zac@ahtlawfirm.com
Larry D. Thompson, Jr.
Texas Bar No. 24051428
larry@ahtlawfirm.com
Christopher Ryan Pinckney
Texas Bar No. 24067819

ryan@ahtlawfirm.com
Rehan M. Safiullah
Texas Bar No. 24066017
rehan@ahtlawfirm.com

ANTONELLI, HARRINGTON
& THOMPSON LLP
4306 Yoakum Blvd., Ste. 450
Houston, TX 77006
(713) 581-3000

Stafford Davis
State Bar No. 24054605
sdavis@stafforddavisfirm.com
Catherine Bartles
Texas Bar No. 24104849
cbartles@stafforddavisfirm.com
THE STAFFORD DAVIS FIRM
815 South Broadway Avenue
Tyler, Texas 75701
(903) 593-7000
(903) 705-7369 fax

Attorneys for Liberty Patents, LLC